BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem yang baru. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman.

4.1. Implementasi Sistem

Implementasi sistem merupakan tahap yang dilakukan setelah melakukan tahap perancangan sistem yaitu menjelaskan mengenai pembuatan sistem serta tampilan dari aplikasi yang telah dibuat, yang sesuai dengan analisis dan perancangan sebelumnya. Setelah tahap implementasi dilakukan maka dibutuhkan sebuah pengujian sistem untuk membuktikan bahwa aplikasi dapat berjalan sesuai dengan yang diharapkan.

4.1.1. Perangkat Keras

Perangkat keras untuk mengimplementasikan sistem adalah sebuah *web server* denganspesifikasi minimal sebagai berikut:

1. *Intel Pentium 3*
2. Memori : 2GB
3. Hardisk : 80GB

4.1.2. Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan sistem adalah sebagai berikut:

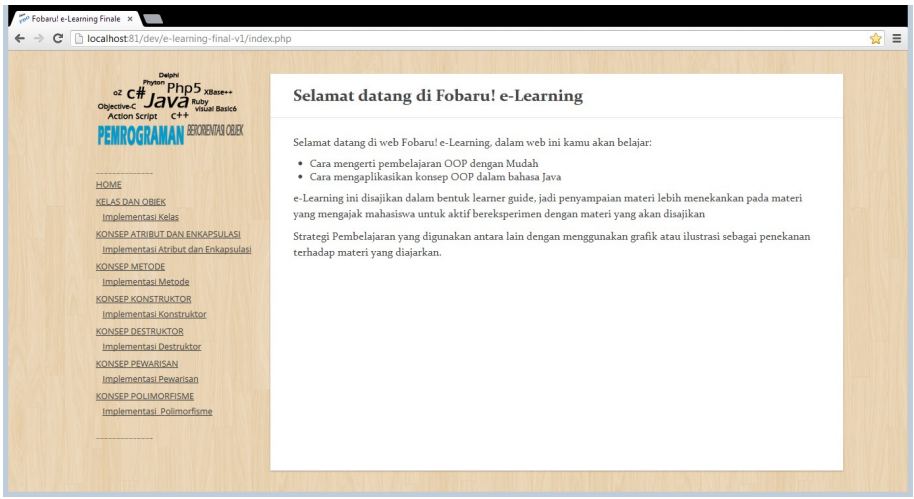
1. Sistem Operasi : Linux/Microsoft Windows Server
2. *Netbeans, Adobe Photoshop*
3. Php5
4. MySql 4
5. Mozilla Firefox, Internet Explorer dan Opera sebagai *web browser*

4.1.3. Implementasi Tampilan Program

Implementasi tampilan program merupakan pemaparan mengenai tampilan-tampilan dari sistem baru yang akan berjalan. Untuk memperjelas bentuk dari implementasi tampilan program, berikut pemaparan dan fungsi dari setiap tampilan yang telah dibuat :

4.1.3.1. Tampilan Halaman Utama

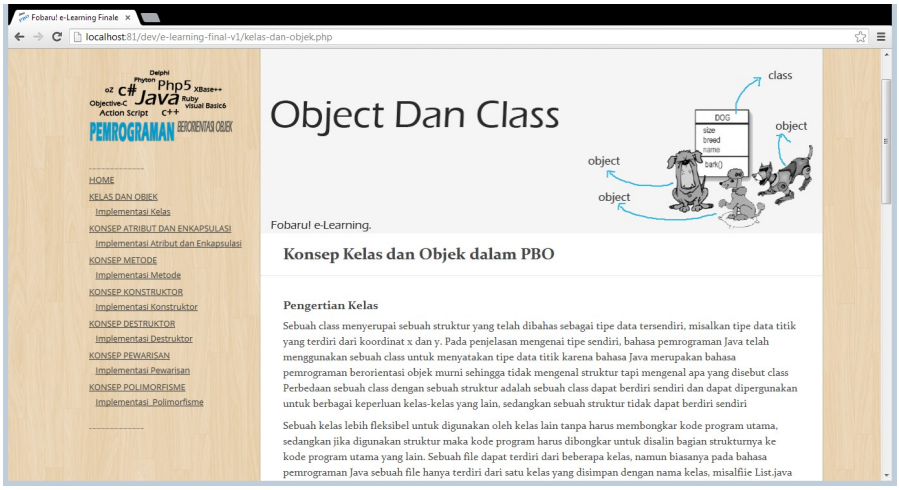
Halaman utama ialah halaman yang pertama kali dibuka, yakni ketika pengguna mengetikan halaman *website* aplikasi *e-learning* untuk bahasa pemrograman berorientasi objek. Pada halaman utama ini memberikan pengenalan tentang aplikasi yang dibangun. Untuk lebih jelasnya bisa dilihat pada gambar 4.1 dibawah ini:



Gambar 4.1. Tampilan Halaman Utama

4.1.3.2. Tampilan Halaman Kelas dan Objek

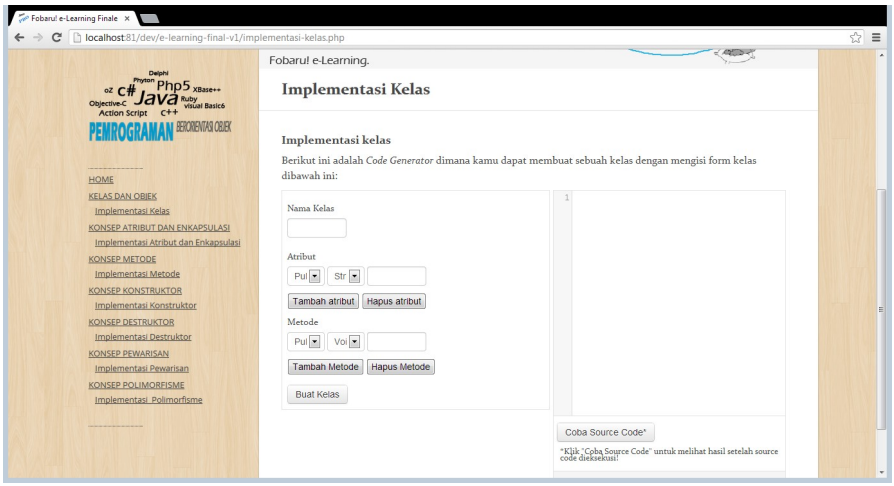
Pada halaman ini dijelaskan tentang konsep kelas dan objek dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.2 dibawah ini:



Gambar 4.2. Tampilan Halaman Kelas dan Objek

4.1.3.3. Tampilan Halaman Implementasi Kelas

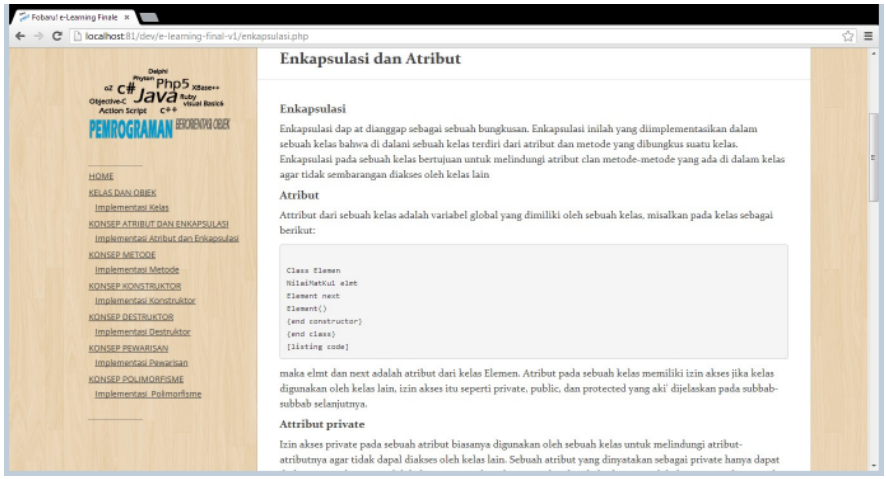
Pada halaman ini terdapat *code generator* untuk membuat kerangka *source code* dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.3 dibawah ini:



Gambar 4.3. Tampilan Halaman Implementasi Kelas

4.1.3.4. Tampilan Halaman Konsep Atribut dan Enkapsulasi

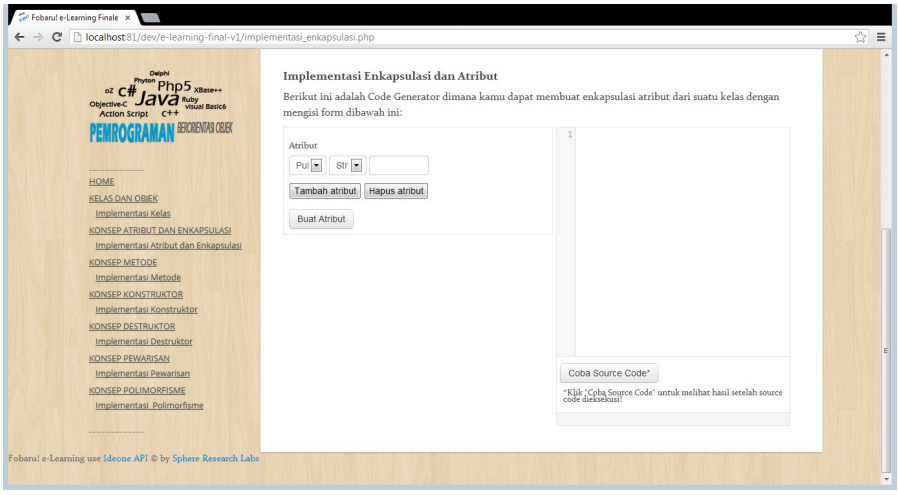
Pada halaman ini dijelaskan tentang konsep atribut dan enkapsulasi dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.4 dibawah ini:



Gambar 4.4. Tampilan Halaman Konsep Atribut dan Enkapsulasi

4.1.3.5. Tampilan Halaman Implementasi Atribut dan Enkapsulasi

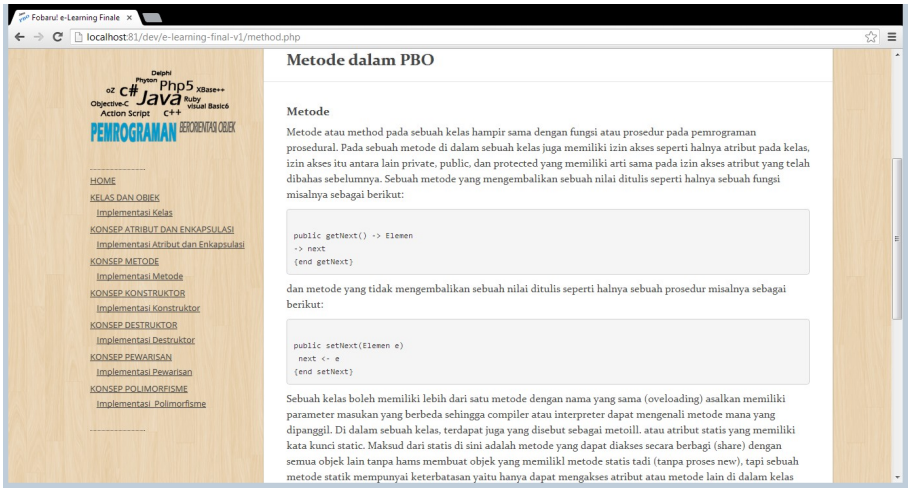
Pada halaman ini terdapat code generator untuk membuat kerangka source code dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.5 dibawah ini:



Gambar 4.5. Tampilan Halaman Implementasi Atribut dan Enkapsulasi

4.1.3.6. Tampilan Halaman Konsep Metode

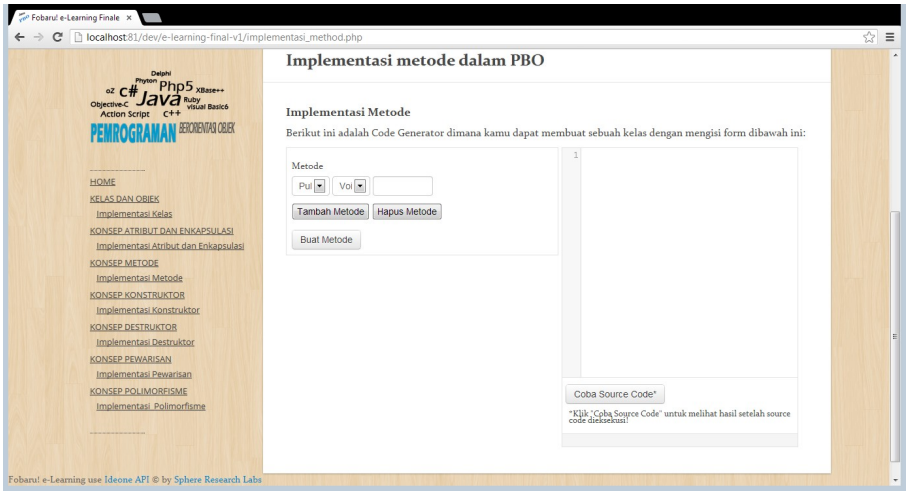
Pada halaman ini dijelaskan tentang konsep metode dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.6 dibawah ini:



Gambar 4.6. Tampilan Halaman Metode

4.1.3.7. Tampilan Halaman Implementasi Metode

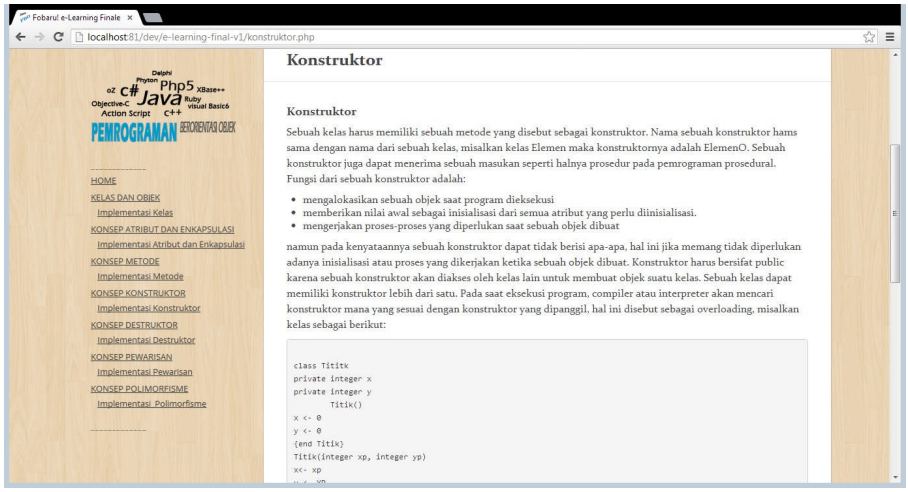
Pada halaman ini terdapat code generator untuk membuat kerangka source code dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.7 dibawah ini:



Gambar 4.7. Tampilan Halaman Implementasi Metode

4.1.3.8. Tampilan Halaman Konsep Konstruktor

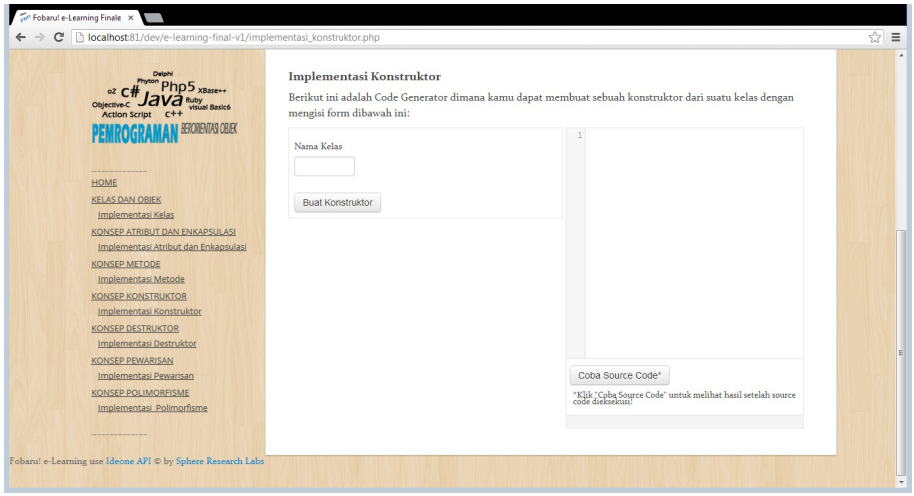
Pada halaman ini dijelaskan tentang konsep konstruktor dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.8 dibawah ini:



Gambar 4.8. Tampilan Halaman Konsep Konstruktor

4.1.3.9. Tampilan Halaman Implementasi Konstruktor

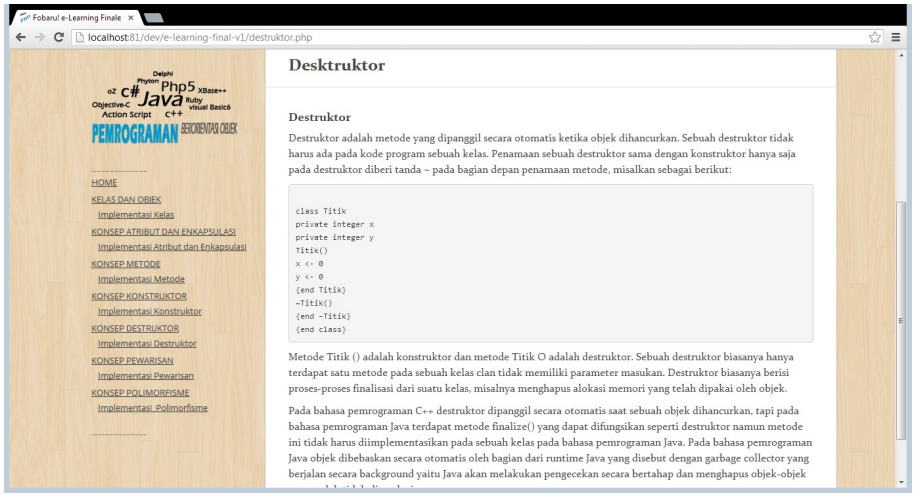
Pada halaman ini terdapat code generator untuk membuat kerangka source code dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.9 dibawah ini:



Gambar 4.9. Tampilan Halaman Implementasi Konstruktor

4.1.3.10. Tampilan Halaman Konsep Destruktor

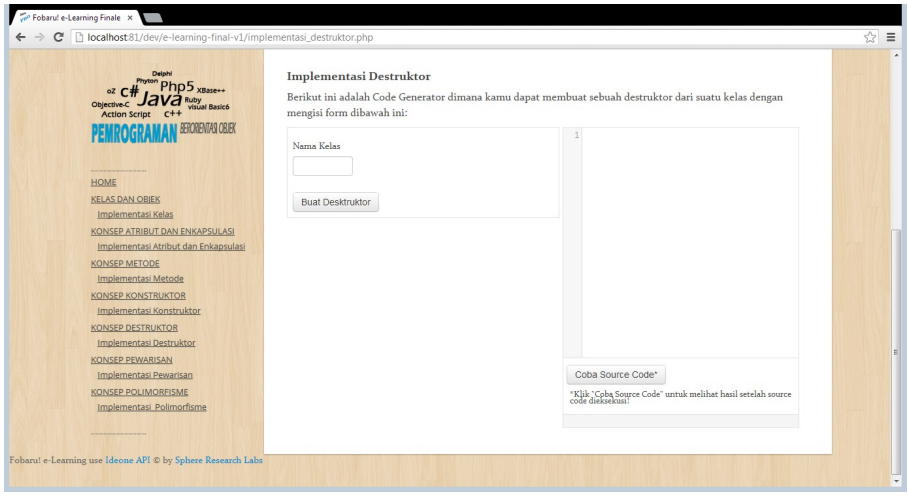
Pada halaman ini dijelaskan tentang konsep destruktor dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.10 dibawah ini:



Gambar 4.10. Tampilan Halaman Konsep Destruktor

4.1.3.11. Tampilan Halaman Implementasi Destruktor

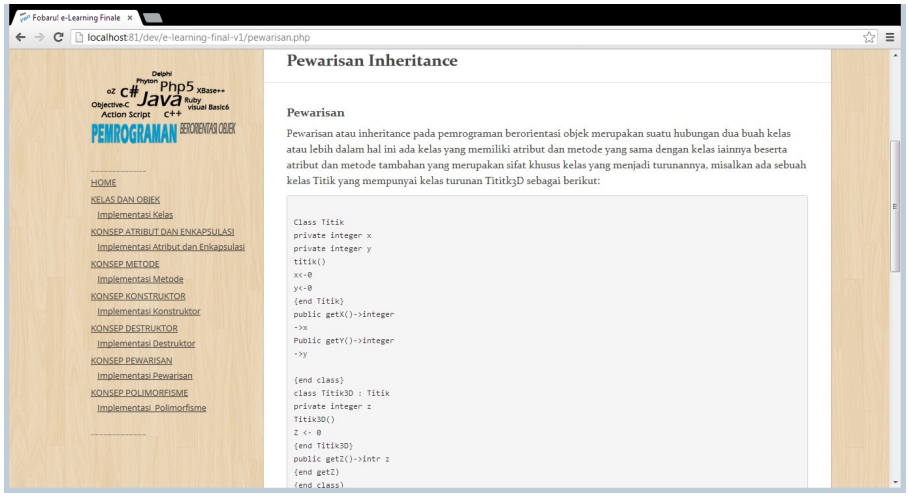
Pada halaman ini terdapat code generator untuk membuat kerangka source code dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.11 dibawah ini:



Gambar 4.11. Tampilan Halaman Implementasi Destruktor

4.1.3.12. Tampilan Halaman Konsep Pewarisan

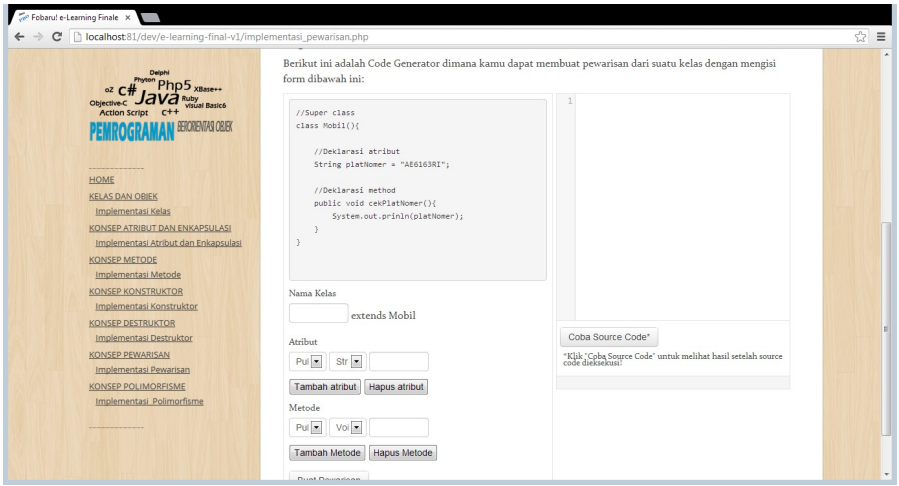
Pada halaman ini dijelaskan tentang konsep pewarisan dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.12 dibawah ini:



Gambar 4.12. Tampilan Halaman Konsep Pewarisan

4.1.3.13. Tampilan Halaman Implementasi Pewarisan

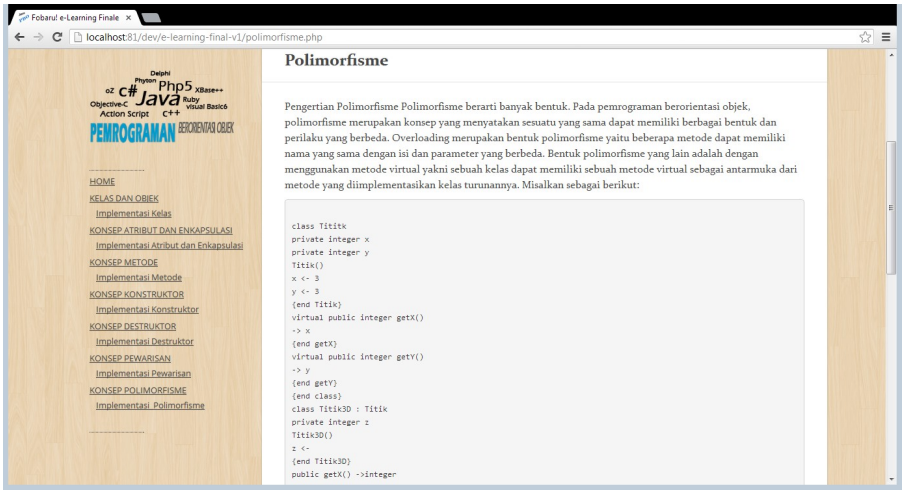
Pada halaman ini terdapat code generator untuk membuat kerangka source code dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.13 dibawah ini:



Gambar 4.13. Tampilan Halaman Implementasi Pewarisan

4.1.3.14. Tampilan Halaman Konsep Polimorfisme

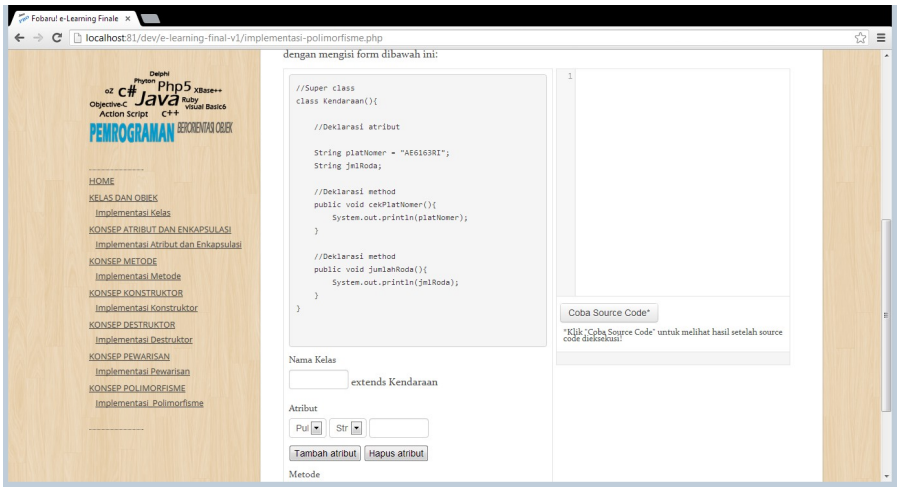
Pada halaman ini dijelaskan tentang konsep metode dalam PBO, untuk lebih jelasnya bisa dilihat pada gambar 4.14 dibawah ini:



Gambar 4.14. Tampilan Halaman Konsep Polimorfisme

4.1.3.15. Tampilan Halaman Implementasi Polimorfisme

Pada halaman ini terdapat code generator untuk membuat kerangka source code dengan mengisikan data pada form yang disediakan, untuk lebih jelasnya dapat dilihat pada gambar 4.15 dibawah ini:



Gambar 4.15. Tampilan Halaman Implementasi Polimorfisme

4.2. Pengujian Sistem

Pengujian yang digunakan untuk menguji sistem yang baru adalah metode pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

4.2.1. Rencana Pengujian

Rencana pengujian akan dilakukan dengan menguji sistem secara *alpha* dan *beta.* Rencana pengujian selengkapnya terlihat pada tabel 4.1 dibawah ini:

Tabel 4.1. Tabel Rencana Pengujian

Item Uji	Detail Pengujian	Jenis Pengujian
Halaman Materi	Tampilkan materi	*Black Box*
	Tampilkan Coba *Source Code*	*Black Box*
	Tampilkan *Code Generator*	*Black Box*

4.2.2. Kasus Dan Hasil Pengujian *Alpha*

Berdasarkan rencana yang telah disusun, maka dapat dilakukan pengujian sebagai berikut:

4.2.2.1. Pengujian Halaman Materi

Hasil pengujian *apha* untuk tampilkan materi dapat dilihat pada tabel 4.2 dibawah ini:

Tabel 4.2. Pengujian Tampilkan Materi

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Id materi	Tercantum *address bar*	Dapat mengisi data sesuai yang diharapkan dan menampilkan data dengan benar	[✓] Diterima [ ] Ditolak

Hasil pengujian *alpha* untuk tampilkan *coba source code* dapat dilihat pada tabel 4.3 dibawah ini:

Tabel 4.3. Pengujian Tampilkan Coba *Source Code*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Source code yang terdapat dalam materi	Tercantum pada halaman materi	Dapat menampilkan informasi sesuai yang diharapkan	[✓] Diterima [ ] Ditolak

Hasil pengujian *alpha* untuk tampilkan *code generator* dapat dilihat pada tabel 4.4 dibawah ini:

Tabel 4.4. Pengujian Tampilkan *Code Generator*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama kelas, nama properties, nama method	Tercantum pada halaman materi	Dapat menampilkan hasil dari code generator seperti yang diharapkan	[✓] Diterima [ ] Ditolak

4.2.3. Kesimpulan Hasil Pengujian *Alpha*

Berdasarkan hasil pengujian *alpha* (fungsional) dengan sample uji yang telah dilakukan dapat ditarik kesimpulan bahwa secara fungsional aplikasi *e-learning* untuk pemrograman berorientasi objek ini tidak terdapat kesalahan dan dapat mengeluarkan hasil sesuai dengan yang diharapkan.

4.2.4. Kasus Dan Hasil Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dimana aplikasi diuji secara langsung ke lapangan yaitu dengan menggunakan kuisioner mengenai kepuasan pengguna (*user*) terhadap aplikasi yang telah dibangun, dan apakah sesuai dengan yang diharapkan. Adapun metode penilaian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari *user.*

4.2.5. Skenario Pengujian *Beta*

Kuisioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 30 mahasiswa yang telah pernah belajar bahasa pemrograman dan setiap responden diberikan kesempatan untuk mencoba aplikasi secara langsung. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem. Kuisioner ini terdapat 9 pertanyaan (kuisioner dapat dilihat pada lampiran).

Adapun sebelum mengisi pertanyaan kuisioner pengujian, setiap responden diberikan enam pertanyaan validitas dan pertanyaan rengalaman responden dalam memperlajari bahasa pemrograman berorientasi objek. Semua responden dinilai layak dijadikan acuan data sampel karena semua responden

adalah mahasiswa yang pernah belajar bahasa pemrograman dan pernah menggunakan IDE seperti *Netbeans* dan *Eclipse.*

Berdasarkan data hasil kuisioner tersebut dicari presentase masing-masing jawaban dengan menggunakan rumus:

Y = P/Q * 100%

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai persentase

4.2.5.1. Hasil Pengujian Kuisioner Nomer 1

Pertanyaan: Apakah anda setuju dengan adanya aplikasi ini membuat anda lebih mudah mempelajari bahasa pemrograman berorientasi objek?

Tabel 4.5. Hasil Pengujian Kuisioner Soal No 1

Pilihan	Responden	Presentase
Setuju	24	80%
Biasa	6	20%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 24 responden atau 80% menyatakan setuju, 2 orang atau 20% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa aplikasi ini mempermudah dalam mempelajari bahasa pemrograman berorientasi objek.

4.2.5.2. Hasil Pengujian Kuisioner Nomer 2

Pertanyaan: Apakah anda setuju dengan adanya aplikasi ini mempermudah anda dalam memahami konsep pemrograman berorientasi objek?

Tabel 4.6. Hasil Pengujian Kuisioner Soal No 2

Pilihan	Responden	Presentase
Setuju	25	83.33%

Biasa	5	16.67%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 25 responden atau 83.33% menyatakan setuju, 5 orang atau 16.67% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa aplikasi ini mempermudah dalam memahami konsep pemrograman berorientasi objek.

4.2.5.3. Hasil Pengujian Kuisioner Nomer 3

Pertanyaan: Apakah anda setuju bahwa dengan adanya aplikasi ini anda lebih mudah menerapkan source code yang ada didalam buku referensi dibandingkan dengan IDE lain seperi Netbeans atau Eclipse?

Tabel 4.7. Hasil Pengujian Kuisioner Soal No 3

Pilihan	Responden	Presentase
Setuju	24	80.00%
Biasa	6	20.00%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 24 responden atau 80.00% menyatakan setuju, 6 orang atau 20.00% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa aplikasi dengan aplikasi ini source code pada buku referensi lebih mudah diterapkan daripada menggunakan IDE lain seperti *Netbeans* atau *Eclipse*.

4.2.5.4. Hasil Pengujian Kuisioner Nomer 4

Pertanyaan: Apakah anda setuju adanya *fitur live compiler* memudahkan anda dalam memahami konsep pemrograman berorientasi objek?

Tabel 4.8. Hasil Pengujian Kuisioner Soal No 4

Pilihan	Responden	Presentase
Setuju	22	73.33%

Biasa	8	26.67%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 22 responden atau 73.33% menyatakan setuju, 8 orang atau 26.67% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa fitur live compiler pada aplikasi ini memudahkan pengguna dalam memahami konsep pemrograman berorientasi objek.

4.2.5.5. Hasil Pengujian Kuisioner Nomer 5

Pertanyaan: Apakah anda setuju danya fitur *code generator* memudahkan anda dalam memahami konsep pemrograman berorientasi objek?

Tabel 4.9. Hasil Pengujian Kuisioner Soal No 5

Pilihan	Responden	Presentase
Setuju	25	83.33%
Biasa	5	16.67%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 25 responden atau 83.33% menyatakan setuju, 5 orang atau 16.67% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa fitur code generator pada aplikasi ini memudahkan dalam memahami konsep pemrograman berorientasi objek.

4.2.5.6. Hasil Pengujian Kuisioner Nomer 6

Pertanyaan: Apakah anda setuju ilustrasi yang digunakan dalam aplikasi ini memudahkan anda dalam memahami konsep bahasa pemrograman berorientasi objek?

Tabel 4.10. Hasil Pengujian Kuisioner Soal No 6

Pilihan	Responden	Presentase
Setuju	25	83.33%
Biasa	5	16.67%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 25 responden atau 83.33% menyatakan setuju, 5 orang atau 16.67% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa ilustrasi yang ada dalam aplikasi ini memudahkan dalam memahami konsep bahasa pemrograman berorientasi objek.

4.2.5.7. Hasil Pengujian Kuisioner Nomer 7

Pertanyaan: Apakah anda setuju materi pembelajaran bahasa pemrograman berorientasi objek yang diberikan mudah dipahami?

Tabel 4.11. Hasil Pengujian Kuisioner Soal No 7

Pilihan	Responden	Presentase
Setuju	24	80.00%
Biasa	6	20.00%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 24 responden atau 80.00% menyatakan setuju, 6 orang atau 20.00% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa materi pembelajaran dalam aplikasi ini mudah dipahami.

4.2.5.8. Hasil Pengujian Kuisioner Nomer 8

Pertanyaan: Apakah anda setuju dengan adanya aplikasi ini berguna bagi anda untuk mempelajari bahasa pemrograman berorientasi objek?

Tabel 4.12. Hasil Pengujian Kuisioner Soal No 8

Pilihan	Responden	Presentase
Setuju	24	80.00%
Biasa	6	20.00%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 24 responden atau 80.00% menyatakan setuju, 6 orang atau 20.00% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan bahwa aplikasi ini berguna untuk memudahkan untuk mempelajari bahasa pemrograman berorientasi objek.

4.2.5.9. Hasil Pengujian Kuisioner Nomer 9

Pertanyaan: Apakah anda setuju antar muka aplikasi ini mudah digunakan?

Tabel 4.13. Hasil Pengujian Kuisioner Soal No 9

Pilihan	Responden	Presentase
Setuju	27	90.00%
Biasa	3	10.00%
Tidak Setuju	0	0%

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 27 responden atau 90.00% menyatakan setuju, 3 orang atau 10.00% menyatakan biasa saja dan tidak ada responden yang menyatakan tidak setuju.

Berdasarkan data diatas dapat disimpulkan antar muka aplikasi ini mudah digunakan.

4.2.6. Kesimpulan Pengujian *Beta*

Dari pengujian *Beta* yang telah dilakukan yaitu dengan pengujian perhitungan pilihan kategori jawaban benar kuisioner yang telah dibagikan di lapangan didapat kesimpulan bahwa aplikasi Learning Management System ini:

1. Aplikasi *e-learning* untuk bahasa pemrograman berorientasi objek ini mempermudah dalam mempelajari bahasa pemrograman berorientasi objek.
1. Aplikasi *e-learning* untuk bahasa pemrograman berorientasi objek ini mempermudah dalam memahami konsep pemrograman berorientasi objek.
2. Lebih mudah diterapkan daripada menggunakan IDE lain seperti *Netbeans* atau *Eclipse.*
3. Fitur *Live Compiler* pada aplikasi ini memudahkan pengguna dalam memahami konsep pemrograman berorientasi objek.
4. Fitur *Code Generator* pada aplikasi ini memudahkan dalam memahami konsep pemrograman berorientasi objek.
5. Ilustrasi yang ada dalam aplikasi ini memudahkan dalam memahami konsep bahasa pemrograman berorientasi objek.
6. Materi pembelajaran dalam aplikasi ini mudah dipahami.
7. Aplikasi *e-learning* untuk bahasa pemrograman berorientasi objek ini berguna untuk memudahkan untuk mempelajari bahasa pemrograman berorientasi objek.
8. Antar muka aplikasi ini mudah digunakan.