SISTEM PENILAIAN OTOMATIS MODUL PEMROGRAMAN JAVA SEDERHANA

Oleh :
RANI PURBANINGTYAS *)

Abstrak

Prototype sistem penilaian otomatis modul pemrograman Java sederhana ini adalah rancangan sistem penilaian otomatis dalam bentuk modul program yang ditujukan untuk meningkatkan efisiensi waktu kerja dalam memeriksa modul program Java yang sifatnya sederhana atau tingkat dasar. Sifat pemeriksaan oleh sistem dibedakan menjadi 2 macam yaitu *white box checking* dan *black box checking* yang disesuaikan dengan kebutuhan pemeriksa modul program Java. Pemeriksaan secara *white box checking* dilakukan sampai dengan pemeriksaan *listing code* modul program Java sederhana menggunakan teknik pencocokan string *regular expression* yang mengimplementasikan *class-class* yang ada dalam *package* java.util.regex.* yaitu *class* Pattern dan *class* Matcher. Keluaran sistem berupa nilai angka yang menunjukkan tingkat kesempurnaan modul program Java yang diperiksa. *Prototype* sistem penilaian otomatis modul pemrograman Java sederhana ini lolos uji *acceptance testing* untuk kategori *functionality* karena telah memenuhi semua kebutuhan *user* (pemeriksa program) terhadap sistem penilaian otomatis. *Prototype* sistem penilaian otomatis ini juga telah lolos uji *acceptance testing* untuk kategori *performance* karena mampu meningkatkan efisiensi waktu yang dibutuhkan untuk memeriksa modul program Java yang sifatnya sederhana atau tingkat dasar.

Kata kunci : sistem penilaian otomatis, *java programming*, *prototype* sistem

PENDAHULUAN

Tugas untuk memeriksa modul program, khususnya program Java, membutuhkan perhatian lebih. Hal ini dikarenakan karakteristik dari bahasa pemrograman Java itu sendiri yang sifatnya *case sensitive*. Belum lagi tuntutan untuk memeriksa logika dari si *programmer*. Selainkan itu, modul program yang dikembangkan menggunakan bahasa pemrograman Java berbeda dengan bahasa pemrograman lain. Hal ini dikarenakan modul program tersebut harus di-*compile* terlebih dahulu, baru bisa dijalankan (di-*run*). Kesalahan dalam penulisan *syntax* program, menyebabkan program tersebut tidak bisa di-*compile*. Disamping itu, apabila profesi pemeriksa itu adalah seorang instruktur program, seperti dosen atau mentor sebuah lembaga kursus, tentu saja membutuhkan waktu yang lebih dari cukup untuk memeriksa modul-modul program yang dibuat oleh anak didiknya. Sehingga peneliti melihat adanya kebutuhan untuk mengembangkan sebuah sistem yang mampu memeriksa dan memberikan penilaian secara otomatis modul program Java sehingga mampu meningkatkan efisiensi waktu kerja dalam memeriksa modul-modul program Java tersebut.

Sebelumnya telah ada penelitian-penelitian sejenis yang juga berupaya membuat sistem penilaian otomatis. Namun, masih terdapat hal-hal yang belum tercapai dari hasil penelitian terdahulu (Halide,2010). Diantaranya adalah tidak dilakukan pengujian input-output dari modul program Java tersebut. Tidak tersedia fasilitas untuk memeriksa *listing code* program sehingga pemeriksa tidak mampu menilai kemampuan bahasa program *programmer* dengan obyektif. Pemberian nilai yang dilakukan oleh sistem bersifat statis sehingga membatasi pemeriksa dalam memberi nilai terhadap modul program Java yang diperiksa. Kunci jawaban berupa *listing code* untuk modul program Java yang diperiksa yang disediakan oleh sistem juga bersifat statis, sehingga tidak bisa diterapkan untuk semua mdul program Java yang akan diperiksa. Sehingga hal ini akan menambah beban pemeriksa karena harus membuat kunci jawaban berupa *listing code* untuk setiap modul program Java yang akan diperiksa.

Berdasarkan hal tersebut diatas, maka penelitian ini bertujuan untuk mengembangkan sebuah sistem yang mampu memeriksa dan memberikan penilaian secara otomatis terhadap modul-modul program Java sehingga mampu meningkatkan efisiensi waktu kerja dalam memeriksa modul-modul program Java tersebut. Karena sistem yang akan dikembangkan ini merupakan sebuah *prototype* awal, maka dikhususkan untuk memeriksa dan memberikan penilaian secara otomatis terhadap modul-modul program Java yang sifatnya

*) Staf Pengajar STMIK Dipanegara Makasar

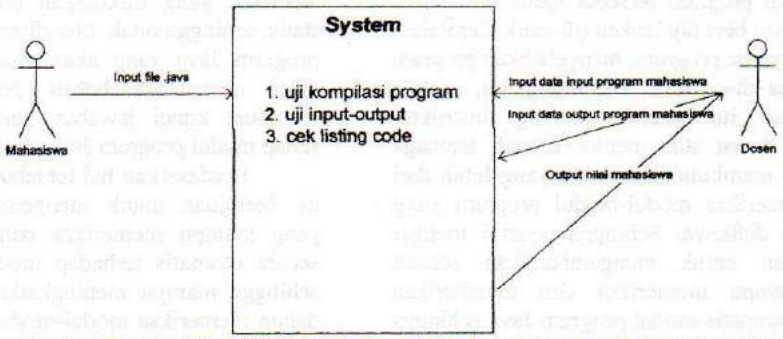
memeriksa modul-modul program Java tersebut. Karena sistem yang akan dikembangkan ini merupakan sebuah *prototype* awal, maka dikhususkan untuk memeriksa dan memberikan penilaian secara otomatis terhadap modul-modul program Java yang sifatnya sederhana atau tingkat dasar. *Prototype* sistem penilaian otomatis ini ditujukan agar dapat mengisi kekurangan dari penelitian-penelitian sebelumnya. Diharapkan dengan menggunakan *prototype* sistem penilaian otomatis ini, waktu yang dibutuhkan untuk memeriksa modul-modul program Java menjadi lebih singkat.

METODE PENELITIAN

Metode pengembangan sistem yang digunakan untuk membuat sistem penilaian otomatis ini menggunakan model prototyping. Model prototyping yang diterapkan dalam pengembangan sistem penilaian otomatis ini dapat digambarkan sebagai berikut :

Gambar 1. Model prototyping sistem penilaian otomatis modul pemrograman Java sederhana

Identifikasi kebutuhan sistem dilakukan di awal kegiatan penelitian. Identifikasi kebutuhan sistem dilakukan dengan cara menanyakan kepada pengguna sistem penilaian otomatis ini, yaitu pemeriksa modul pemrograman Java, tentang hal-hal apa saja yang dibutuhkan untuk memberikan penilaian modul program Java secara otomatis sehingga bisa didapatkan hasil penilaian yang bersifat obyektif. Yang menjadi pemeriksa modul pemrograman Java ini adalah instruktur program Java, seperti dosen atau mentor sebuah lembaga kursus bahasa pemrograman Java.



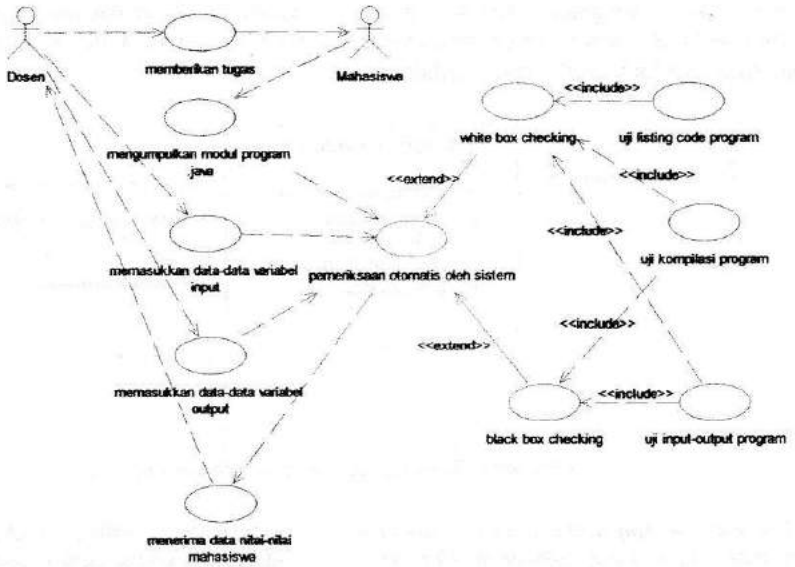
Gambar 2. Identifikasi kebutuhan *user* terhadap sistem penilaian otomatis

Hasil analisis identifikasi kebutuhan user terhadap sistem adalah sebagai berikut:

1. *User* (misal : dosen) memberikan tugas kepada *programmer* (misal : mahasiswa) untuk membuat modul program dalam bahasa Java yang ditujukan untuk menyelesaikan studi kasus tertentu. Mahasiswa akan mengumpulkan tugas tersebut kedalam sistem dalam bentuk file berekstensi .java.

1. *User* (misal : dosen) memberikan tugas kepada *programmer* (misal : mahasiswa) untuk membuat modul program dalam bahasa Java yang ditujukan untuk menyelesaikan studi kasus tertentu. Mahasiswa akan mengumpulkan tugas tersebut kedalam sistem dalam bentuk file berekstensi .java.
2. *User* (dosen) dapat menggunakan *prototype* sistem untuk memeriksa tugas bahasa program Java yang telah berada dalam sistem dengan cara, hanya perlu 1 kali memasukkan nilai input dan nilai output yang akan digunakan sebagai acuan oleh sistem untuk memeriksa output dari keseluruhan program Java yang dibuat oleh *programmer* (mahasiswa).
3. *User* (dosen) mengharapkan sistem dapat menilai logika program mahasiswa, selaku *programmer* Java, dengan cara melakukan pengujian input-output terhadap modul program Java yang telah dikumpulkan oleh mahasiswa ke dalam sistem.
4. *User* (dosen) mengharapkan sistem dapat menilai kemampuan bahasa pemrograman mahasiswa secara cepat dan obyektif dengan cara melakukan pemeriksaan *listing code* modul program Java yang dikumpulkan oleh mahasiswa ke dalam sistem.
5. *User* (dosen) mengharapkan keluaran dari sistem penilaian otomatis ini, yang merupakan hasil pemeriksaan tingkat kesempurnaan modul program Java mahasiswa oleh sistem, dalam bentuk nilai angka.

Hasil analisis identifikasi kebutuhan user terhadap sistem penilaian otomatis jika digambarkan dalam *use case diagram* adalah sebagai berikut :



Gambar 3. *Use case diagram* hasil analisis identifikasi kebutuhan *user* terhadap sistem

Kemudian berdasarkan hasil identifikasi kebutuhan sistem diatas akan dibuat *prototype* dari sistem penilaian otomatis modul pemrograman Java sederhana. *Prototype* sistem penilaian otomatis modul pemrograman Java ini dikembangkan agar dapat berjalan pada *platform* sistem operasi Windows. Untuk memodelkan sistem digunakan pendekatan pemodelan sistem berorientasi obyek dengan mengimplementasikan *Unified Modelling Language* (UML) menggunakan software Rational Rose 2000 Enterprise Edition. Bahasa pemrograman yang digunakan untuk mengembangkan *prototype* sistem penilaian otomatis modul pemrograman Java sederhana ini yaitu bahasa pemrograman Java. *Software* Java yang digunakan yaitu *Java™ Platform Standard Edition 6 JDK versi 1.6.0*. Sedangkan untuk mengembangkan antar muka *prototype* sistem penilaian otomatis ini digunakan software NetBeans versi 6.0.

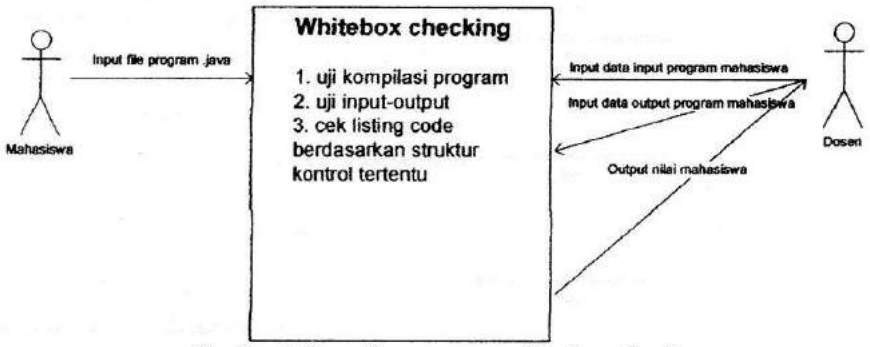
Prototype sistem penilaian otomatis ini terdiri dari beberapa folder yang disesuaikan dengan jumlah penugasan yang diberikan oleh *user*. Dimana pada setiap folder akan berisi file input dan output yang dibuat oleh *user* melalui sistem, yang berisi data-data input dan output yang akan digunakan sebagai acuan oleh sistem untuk memeriksa modul program Java. Folder tersebut juga berisi file-file modul program Java yang akan diperiksa, yang

juga berisi file-file modul program Java yang akan diperiksa, yang disimpan ke dalam sistem dengan menggunakan ekstensi file .java. Hasil pemeriksaan oleh sistem berupa nilai angka juga akan disimpan dalam file tersendiri yang berekstensi .txt.

Sistem akan melakukan tiga tahap pemeriksaan secara berurutan terhadap setiap modul-modul program Java yang ada dalam sistem sebagai berikut :

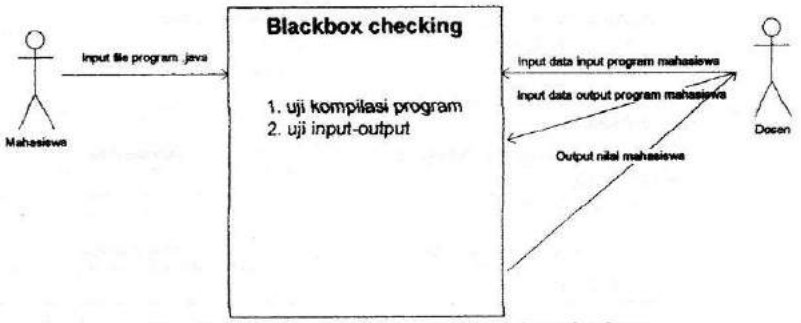
1. Tahap pemeriksaan pertama yaitu eksekusi program. Sistem akan melakukan eksekusi terhadap modul-modul program Java yang ada dalam sistem dengan cara mengkombinasikan beberapa teknik kompilasi untuk menjalankan modul-modul program Java tersebut. Jika modul program Java yang diperiksa dapat dieksekusi, maka sistem akan melanjutkan ke tahap pemeriksaan yang selanjutnya.
2. Tahap pemeriksaan yang kedua yaitu pemeriksaan logika program. Sistem akan memberikan nilai inputan sesuai dengan nilai input yang telah dimasukkan oleh *user* kedalam sistem. Kemudian sistem akan memeriksa apakah modul-modul program Java yang diperiksa tersebut mampu menghasilkan output sama seperti yang telah ditentukan oleh *user*. Jika modul-modul program Java yang ada dalam sistem lolos di tahap ini, maka sistem akan melanjutkan pemeriksaan ke tahap selanjutnya.
3. Tahap pemeriksaan yang ketiga yaitu pemeriksaan *listing code*. Pada tahap ini sistem akan memeriksa apakah *listing code* yang ada dalam modul-modul program Java tersebut sesuai dengan yang diminta oleh *user*. Pada tahap ini tidak semua bagian *listing code* yang ada dalam modul-modul program Java akan diperiksa. Pemeriksaan *listing code* pada tahap ini akan langsung merujuk pada struktur kontrol yang terdalam dalam *listing code* modul program Java tersebut.

Tahapan pemeriksaan oleh sistem seperti yang telah dijelaskan diatas nantinya akan disesuaikan dengan kebutuhan *user*. Sifat pemeriksaan oleh sistem dibedakan menjadi 2 jenis yaitu *white box checking* dan *black box checking*.



Gambar 4. Pemeriksaan secara *white box checking*

White box checking merupakan pemeriksaan modul program Java yang dilakukan jika *user* menginginkan modul program Java yang akan diperiksa tersebut diharuskan untuk merujuk pada struktur kontrol tertentu. Pemeriksaan modul program Java oleh sistem secara *white box checking* akan melalui tahap pemeriksaan eksekusi program, tahap pemeriksaan logika program dengan cara melakukan pengujian input-output, dan tahap pemeriksaan *listing code*. Metodologi yang diterapkan untuk pemeriksaan *listing code* modul program Java menggunakan teknik pencocokan string (*string matching*) yaitu *regular expression*. Metode *regular expression* yang digunakan mengimplementasikan *class-class* yang ada dalam *package* java.util.regex.* yaitu *class* Pattern dan *class* Matcher.



Gambar 5. Pemeriksaan secara *black box checking*

Black box checking merupakan pemeriksaan modul program Java yang dilakukan jika *user* tidak mensyaratkan modul program Java tersebut harus merujuk pada struktur kontrol tertentu dan lebih menekankan pada uji logika program. Pemeriksaan modul program Java oleh sistem secara *black box checking* hanya akan melalui 2 tahap yaitu tahap pemeriksaan eksekusi program dan tahap pemeriksaan logika program dengan cara melakukan pengujian input-output.

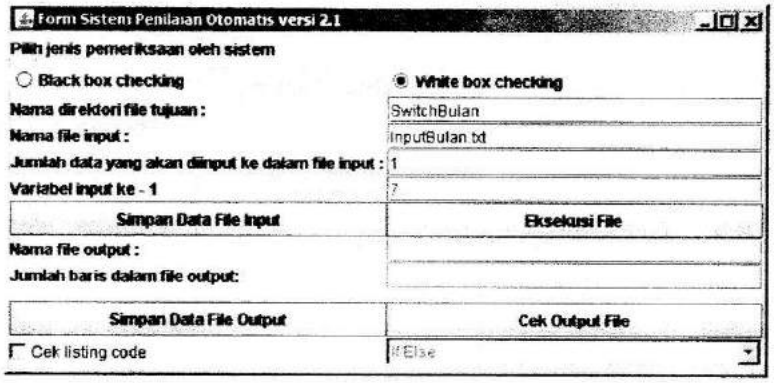
Selanjutnya, prototype sistem penilaian otomatis tersebut akan diimplementasikan. Jika selama tahap implementasi *prototype* sistem penilaian otomatis ini ditemukan permasalahan atau didapatkan kebutuhan yang baru maka *prototype* sistem penilaian otomatis modul pemrograman Java sederhana ini akan dilakukan revisi. Proses ini akan berulang secara terus menerus hingga didapatkan sistem penilaian otomatis modul pemrograman Java yang sempurna.

HASIL PEMBAHASAN

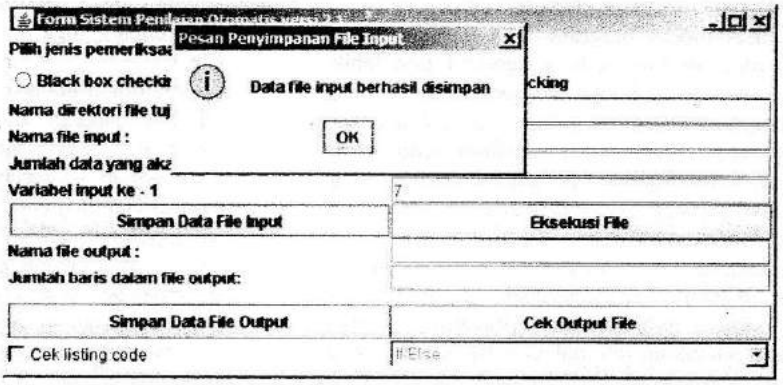
Berikut adalah beberapa persyaratan yang harus dipenuhi agar sistem penilaian otomatis ini dapat berjalan dengan baik selama tahap implementasi :

1. Penamaan *file class* utama modul program Java yang akan diperiksa menggunakan penanda nomor yang *unique* dengan format Pxxxxx sepanjang 11 karakter dan berekstensi .java
2. *File output* modul program Java yang digunakan untuk menyimpan keluaran atau *output* dari modul program Java tersebut disimpan menggunakan nama file yang sama dengan file *class* utama modal program Java tersebut dengan format Pxxxxx sepanjang 11 karakter dan berekstensi .txt.
3. *Programmer* yang akan diperiksa modul program Java buatannya harus diberi materi tentang operasi file terlebih dahulu. Hal ini dikarenakan untuk semua inputan yang akan digunakan untuk menjalankan modul program Java milik *programmer* tersebut berasal dari file inputan milik *user* dan output program milik *programmer* juga harus disimpan dalam bentuk file.
4. *User* diminta untuk menjelaskan terlebih dahulu isi file inputan milik *user*. Hal ini untuk mencegah *programmer* modul program Java yang akan diperiksa melakukan kesalahan penulisan *listing code* untuk mengambil data dari file inputan *user*.
5. Jika *user* menginginkan sistem melakukan pemeriksaan secara *white box checking* maka perlu disarankan kepada *programmer* modul program Java untuk menuliskan *listing code* pengambilan data dari file inputan *user* ditulis dalam *file class* tersendiri. Hal ini bertujuan untuk meminimalkan kesalahan sistem pada saat melakukan pemeriksaan *listing code* berdasarkan struktur kontrol tertentu.
6. Hindari penamaan *file class* selain *file class* Java yang utama menggunakan huruf awal "P".

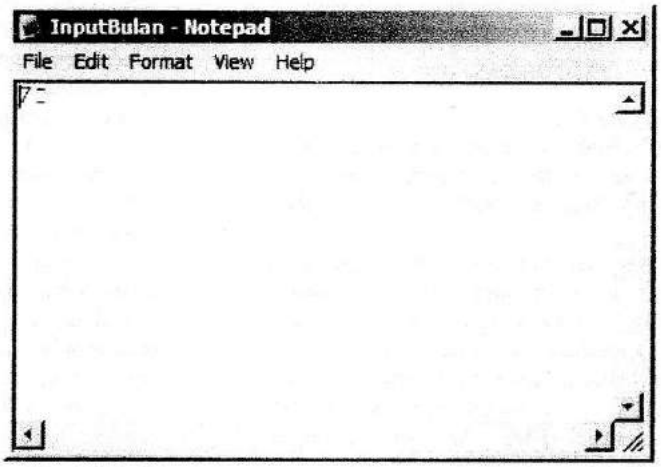
Untuk memperjelas penjelasan tentang implementasi *prototype* sistem penilaian otomatis modul pemrograman Java sederhana ini akan ditampilkan *layout* sistem menggunakan contoh studi kasus pemeriksaan modul program Java secara *white box checking*. *User* memberikan tugas kepada *programmer* membuat modul program Java yang digunakan untuk menampilkan nama-nama bulan jika inputan untuk program tersebut berupa angka dengan menggunakan struktur kontrol "switch". Misal, inputan untuk modul program Java nantinya adalah 7 maka keluaran dari modul program Java yang benar adalah "Juli". Jumlah *programmer* yang membuat modul program Java tersebut ada 10 orang yang masing-masing ditandai dengan nomor pengenal P2700208031 hingga P2700208040. Berikut adalah tampilan sistem pada saat melakukan pemeriksaan secara *white box checking* :



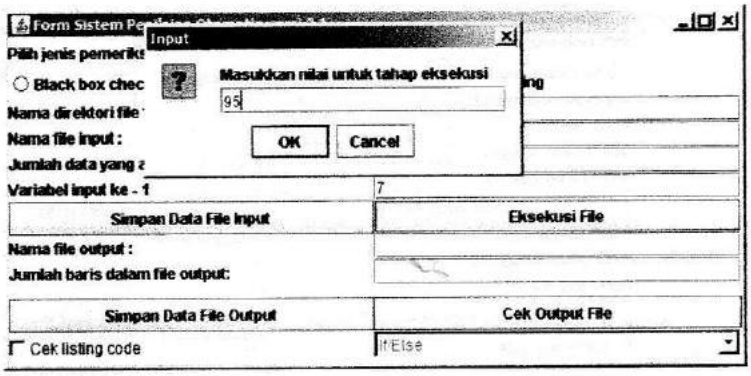
Gambar 6. Tampilan sistem saat *user* memilih pemeriksaan secara *white box checking* dan mengisi file inputan *user*



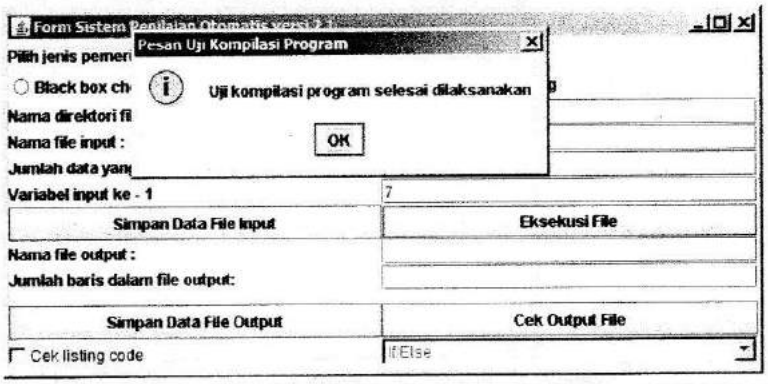
Gambar 7. Pesan dari sistem bahwa berhasil menyimpan data ke file inputan *user*



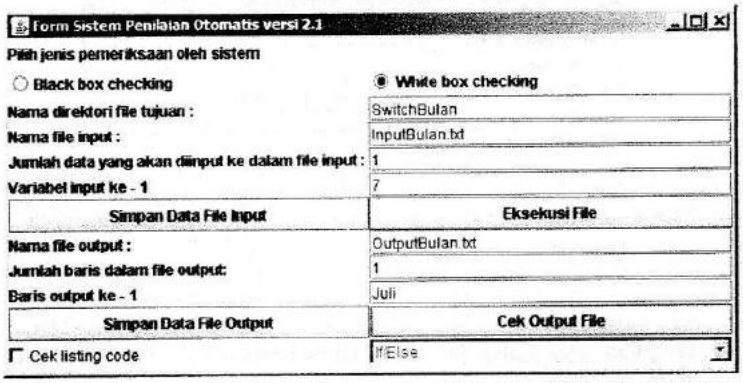
Gambar 8. Isi dari file inputan *user* dengan nama file InputBulan.txt



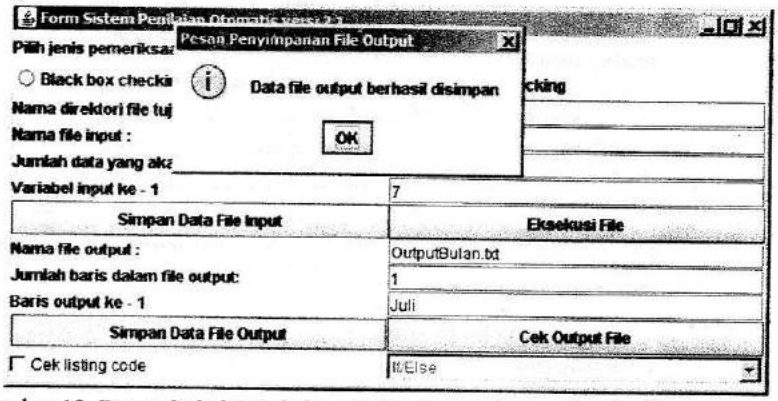
Gambar 9. Tampilan kotak dialog dari sistem untuk inputan nilai modul program Java yang berhasil lolos uji kompilasi program



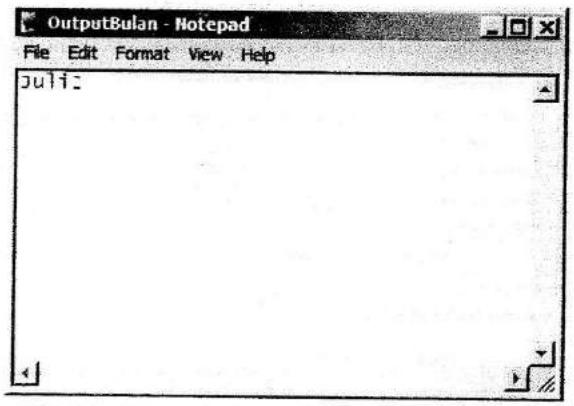
Gambar 10. Pesan dari sistem bahwa sistem telah selesai melaksanakan uji kompilasi progam untuk semua modul program Java yang ada dalam sistem



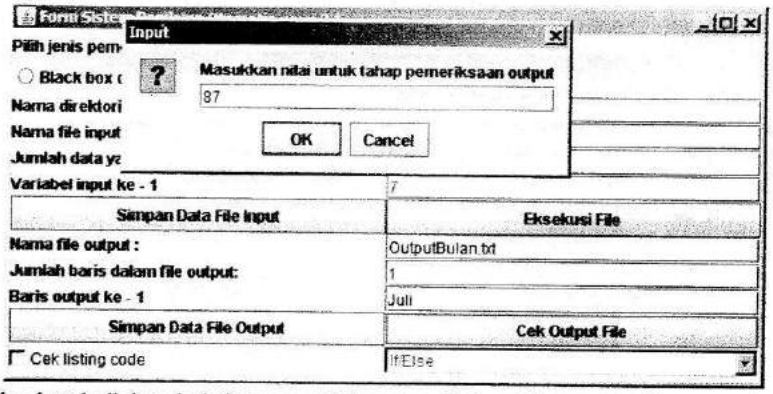
Gambar 11. *User* mengisi data-data yang akan disimpan dalam file output *user*



Gambar 12. Pesan dari sistem bahwa berhasil menyimpan data ke file output *user*

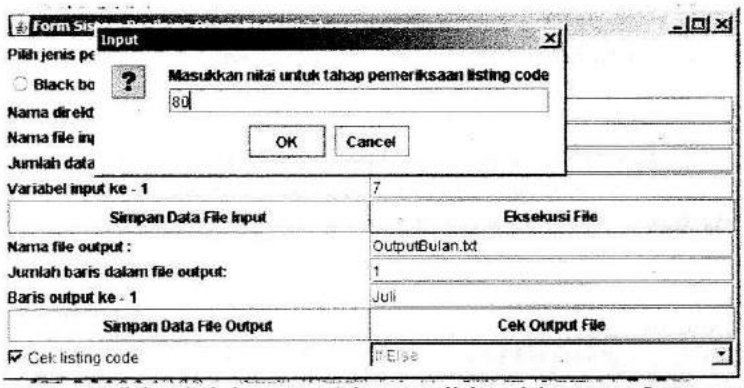


Gambar 13. Isi dari file output *user* dengan nama file OutputBulan.txt

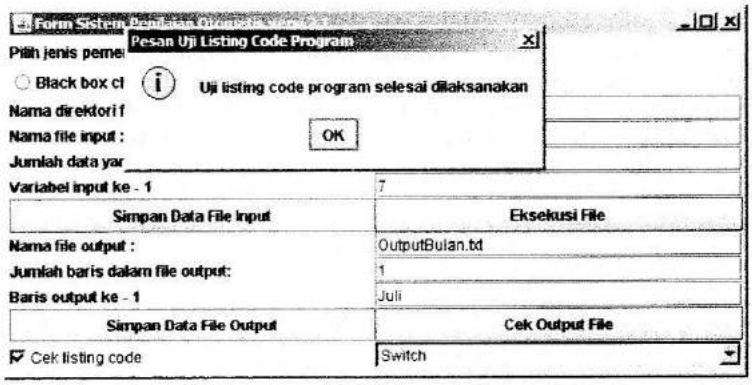


Gambar 14. Tampilan kotak dialog dari sistem untuk inputan nilai modul program Java yang berhasil lolos uji input-output program

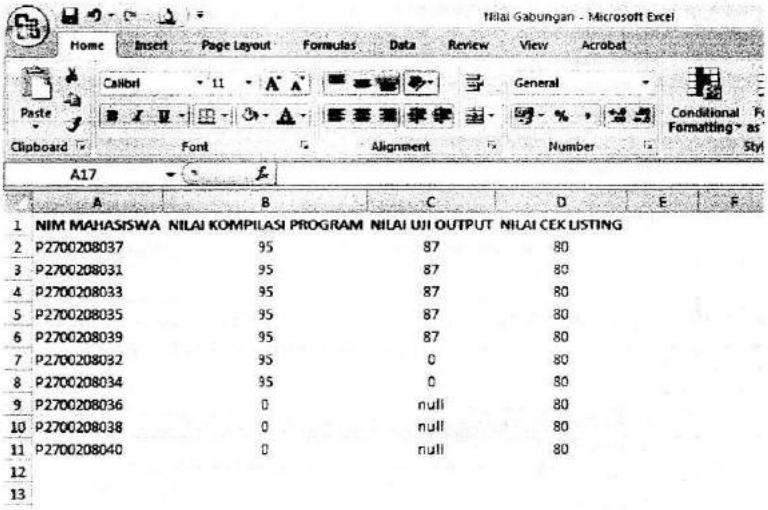
Gambar 15. Pesan dari sistem bahwa sistem telah selesai melaksanakan uji input-output progam untuk semua modul program Java yang ada dalam sistem



Gambar 16. Tampilan kotak dialog dari sistem untuk inputan nilai modul program Java yang berhasil lolos uji *listing code* program



Gambar 17. Pesan dari sistem bahwa sistem telah selesai melaksanakan uji *listing code* progam untuk semua modul program Java yang ada dalam sistem



	A	B	C	D	E	F
1	NIM MAHASISWA	NILAI KOMPILASI PROGRAM	NILAI UJI OUTPUT	NILAI CEK LISTING		
2	P2700208037	95	87	80		
3	P2700208031	95	87	80		
4	P2700208033	95	87	80		
5	P2700208035	95	87	80		
6	P2700208039	95	87	80		
7	P2700208032	95	0	80		
8	P2700208034	95	0	80		
9	P2700208036	0	null	80		
10	P2700208038	0	null	80		
11	P2700208040	0	null	80		
12						
13						

Gambar 18. Tampilan keluaran dari sistem berupa nilai-nilai *programmer* (misal : mahasiswa) dalam bentuk .txt yang dibuka menggunakan aplikasi Ms. Excel 2007

PENGUJIAN PROTOTYPE SISTEM PENILAIAN OTOMATIS

Pengujian *prototype* sistem penilaian otomatis mata kuliah pemrograman Java ini menggunakan teknik *acceptance testing* dengan menitikberatkan pada kategori *functionality* dan *performance*. *Acceptance testing* merupakan salah satu model testing yang digunakan untuk mengetahui apakah *user* menerima sistem tersebut atau tidak berdasarkan kategori tertentu [10].

Functionality merupakan kategori penerimaan *user* terhadap sistem ditinjau dari sisi terpenuhinya atau tercapainya kebutuhan *user*. Faktor-faktor yang mendukung terpenuhinya kategori *functionality* ditentukan berdasarkan identifikasi kebutuhan *user* terhadap sistem yang dibuat. *Performance* merupakan kategori penerimaan *user* terhadap sistem dilihat dari sisi efisiensi pemanfaatan sistem untuk memenuhi kebutuhan *user*. Terpenuhi tidaknya kategori *performance* dilakukan dengan cara melakukan perbandingan perhitungan waktu kerja *user* dengan dan tanpa memanfaatkan sistem.

Hasil pengujian *prototype* sistem penilaian otomatis modul pemrograman Java menggunakan teknik *acceptance testing* untuk kategori *functionality* adalah sebagai berikut :

Tabel 1. Hasil *acceptance testing* untuk kategori *functionality*

IDENTIFIKASI KEBUTUHAN	TERPENUHI
File modul program Java yang dikumpulkan berekstensi .java	√
Memasukkan 1 kali data-data input untuk semua modul program Java yang akan diperiksa	√
Memasukkan 1 kali data-data output untuk semua modul program Java yang akan diperiksa	√
Dilakukan pengujian input-ouput	√
Dilakukan pengujian *listing code*	√
Keluaran sistem berupa nilai angka	√

Dari hasil pengujian diatas dapat disimpulkan bahwa *prototype* sistem penilaian otomatis modul pemrograman Java ini lolos uji *acceptance testing* untuk kategori *functionality* karena telah memenuhi

Tabel 2. Hasil *acceptance testing* untuk kategori *performance*

CONTOH TUGAS	WAKTU KERJA SECARA MANUAL				WAKTU KERJA SISTEM
	RESPONDEN 1	RESPONDEN 2	RESPONDEN 3	RATA-RATA	
Menghitung gaji pegawai	00:11:55	00:05:00	00:20:00	00:12:18	00:01:00
Konversi suhu	00:05:00	00:10:00	00:45:00	00:20:00	00:01:00
Menampilkan nama bulan	00:10:00	00:05:00	00:31:10	00:15:00	00:00:45
Menghitung nilai rata-rata suatu deret bilangan	00:07:30	00:05:20	00:18:05	00:10:18	00:01:00

Dari hasil pengujian diatas dapat dilihat bahwa *prototype* sistem penilaian otomatis ini membutuhkan waktu yang lebih singkat untuk memeriksa modul program Java dibandingkan dengan waktu yang yang dibutuhkan untuk memeriksa modul program Java secara manual. Sehingga dapat disimpulkan bahwa *prototype* sistem penilaian otomatis modul pemrograman Java ini lolos uji *acceptance testing* untuk kategori *performance* karena mampu meningkatkan efisiensi waktu yang dibutuhkan oleh responden untuk memeriksa modul program Java yang sifatnya sederhana.

KESIMPULAN DAN SARAN

Prototype sistem penilaian otomatis modul pemrograman Java sederhana ini merupakan sebuah *prototype* sistem yang mampu memeriksa dan memberikan penilaian secara otomatis terhadap modul-modul program Java yang sifatnya sederhana atau tingkat dasar. Dari hasil pengujian dapat disimpulkan bahwa *prototype* sistem penilaian otomatis modul pemrograman Java ini lolos uji *acceptance testing* untuk kategori *functionality* karena telah memenuhi semua kebutuhan *user* terhadap sistem penilaian otomatis. Dan juga telah lolos uji *acceptance testing* untuk kategori *performance* karena mampu meningkatkan efisiensi waktu yang dibutuhkan oleh responden untuk memeriksa modul program Java yang sifatnya sederhana.

Berikut adalah hal-hal yang disarankan untuk pengembangan prototype sistem penilaian otomatis ini agar didapat sistem penilaian otomatis pemrograman Java yang sempurna :

1. Perlu dikembangkan referensi atau *library* yang dapat mengakomodasi pemberian nilai input terhadap program Java yang akan diperiksa sehingga programmer tidak perlu mempelajari operasi file terlebih dahulu untuk pengerjaan tugas modul program Java.
2. Prototype sistem penilaian otomatis ini perlu dikembangkan agar dapat digunakan untuk memeriksa modul bahasa pemrograman lain selain Java.

DAFTAR PUSTAKA

Charras, Christian, Thierry Lecroq. *HandBook of String Matching*. Algorithms. http://www-igm.univ-mlv.fr/~lecroq/string/string.pdf, diakses tanggal : 24 Desember 2009.

Dietel, R.J, J.L. Herman, R.A. Knuth. 1991. *What Does Research Say About Assessment*, http://methodenpool.uni-koeln.de/portfolio/WhatDoesResearchSayAboutAssessment.htm. North Central Regional Educational Laboratory (NCREL). Oak Brook, diakses tanggal : 29 Desember 2009.

Halide, Lidemar. Tesis. *Automatic Assessment Untuk Mata Kuliah Bahasa Pemrograman Pada Sistem Pembelajaran Berbasis Web*. Program Pascasarjana Universitas Hasanuddin Makassar. 2010.

J.E.N.I. (Java Education Network Indonesia). *Pengenalan Pemrograman Java 1*.

Kumara, Gozali Harda. Tugas Akhir. *Visualisasi Beberapa Algoritma Pencocokan String Dengan Java*. Sekolah Teknik Elektro Informatika. Institut Teknologi Bandung.

Liang,Y.Daniel.2004. *Introduction to Java Programming 6th edition : Comprehensive Version*. Prentice Hall. Amerika Serikat.

Mertz, David. *Introduction to The Tutorial*. http://gnosis.cx/publish/programming/regular_expressions.html, diakses tanggal : 14 Agustus 2010.

Perry, William. 1995. *Effective Methods for Software Testing*. John Wiley & Sons, Inc. Kanada.