

# PERANCANGAN APLIKASI PEMERIKSAAN LISTING PROGRAM PHP PADA E-LEARNING

Annah & Muhajirin

Sistem Informasi, STMIK Dipanegara & Teknik Informatik, STMIK AKBA

[anna.dsndp@gmail.com](mailto:anna.dsndp@gmail.com), [adjieq@gmail.com](mailto:adjieq@gmail.com)

## ABSTRAK

Penelitian ini bertujuan untuk membangun suatu sistem yang dapat memeriksa tugas mahasiswa yang menggunakan bahasa pemrograman PHP dalam e-learning. Efisiensi waktu bagi dosen dalam memeriksa tugas mahasiswa begitu pula bagi mahasiswa dapat langsung melihat nilainya pada saat itu juga sehingga transparansi nilai dapat dijamin. Pengujian sistem dalam penelitian ini digunakan metode eksperimental. Mahasiswa diberikan tugas melalui e-learning kemudian mahasiswa mengirim jawaban dalam bentuk listing program dan pada saat dinyatakan terkirim maka sistem langsung memeriksa jawaban mahasiswa tersebut. Pengiriman jawaban untuk satu tugas hanya dapat dilakukan satu kali. Eksperimen dilakukan terhadap tugas mahasiswa yang terdiri dari 20 mahasiswa. Keluaran sistem dalam bentuk angka dalam hal ini nilai mahasiswa. Hasil penelitian ini menunjukkan bahwa sistem dapat memeriksa listing program yang menggunakan bahasa pemrograman PHP dalam kasus apapun. Jika ditinjau dari segi kecepatan pemeriksaan maka dapat disimpulkan bahwa pemeriksaan listing program dengan menggunakan sistem ini jauh lebih cepat dibandingkan dengan memeriksa secara manual dan keakuratan hasil pemeriksaan juga dapat lebih meningkat serta nilai mahasiswa juga dapat lebih transparan.

*Kata kunci: Bahasa Pemrograman PHP, Pemeriksaan listing program, e-learning*

## A. Pendahuluan

Sistem pembelajaran yang diterapkan oleh sebagian sekolah tinggi maupun universitas saat sekarang ini sudah bersifat elektronik atau yang biasa disebut *e-learning*. Sistem pembelajaran elektronik konsepnya adalah menyediakan pembelajaran dengan menggunakan jaringan internet yang mana peserta didik dan dosen/guru sebagai user dan di dalamnya itu terdapat sejumlah materi pelajaran beserta tugas/soal ujian. Selain itu peserta didik dapat berinteraksi dengan tutor jika ada materi yang kurang dipahami.

Umumnya *e-learning* menyediakan beberapa kemudahan-kemudahan (free dan

dapat dimodifikasi oleh siapapun) serta menyediakan berbagai fungsi-fungsi. Salah satu fungsinya adalah *Course Management* yang di dalamnya terdapat beberapa modul diantaranya *Assignment*. Fasilitas *Assignment*, dosen/guru dimungkinkan untuk memberikan soal tugas/ujian ke mahasiswa/siswa dengan mengharuskan mahasiswa/siswa mengirim kembali jawabannya. Semua kategori soal dapat disajikan di moodle. Pemeriksaan jawaban otomatis untuk kategori soal *true-false*, isian dan *multiple choice*, *essay* juga sudah diterapkan namun untuk soal *essay* yang berbasis bahasa pemrograman belum disediakan.

Kesulitan yang dihadapi ketika dosen memeriksa jawaban dari soal yang diberikan kepada mahasiswa adalah jawaban soal bahasa pemrograman dapat bervariasi. Untuk kelas yang jumlah mahasiswanya sedikit masih memungkinkan dilakukan secara manual namun untuk kelas yang jumlah mahasiswanya besar membutuhkan waktu yang cukup besar pula. Oleh sebab itu sistem pemeriksa *source code* otomatis tentunya sangat membantu.

Sistem penilai otomatis telah dibangun oleh beberapa peneliti terdahulu diantaranya system menilai *source code* dengan *input-outputnya* di mana mahasiswa diharuskan menggunakan kode operasi yang telah ditentukan oleh dosen/guru pengampu sehingga mahasiswa tidak dapat berkreasi. Bahasa pemrograman yang dinilai adalah Java (Rani Purbaningtyas, 2010). Sedangkan penelitian berikutnya telah mengembangkan penilai *source code* otomatis bersifat generik dalam bahasa pemrograman Pascal dan Java. Peneliti ini menilai *source code* dari segi sintaks, leksikal, dan flowgraphnya namun belum diintegrasikan dengan *e-learning* sehingga dosen/guru harus mendownload jawaban kemudian dimasukkan ke sistem pemeriksa otomatis. Hal ini juga tentunya membutuhkan waktu yang cukup lama. Soal tugas/ujian yang diberikan kepada mahasiswa juga masih secara manual (Meylanie Olivya, 2011).

Dari sistem yang dibangun oleh peneliti terakhir akan dikembangkan dengan mengimplementasikan pembelajaran elektronik bahasa pemrograman PHP dengan sistem penilai *source code* otomatis sebagai plugin moodle. Dengan mengintegrasikan dengan moodle maka diharapkan dosen/ guru dan mahasiswa/siswa sebagai user memperoleh kemudahan mulai dari penyajian soal-soal sampai penilaian tugas/ujian bahasa pemrograman PHP (Muchammad Nur Qomaruddin, 2009).

Tujuan penelitian ini adalah untuk merancang sistem pemeriksa *source code* otomatis yang dapat memeriksa dan menilai program yang menggunakan bahasa pemrograman PHP serta untuk mengintegrasikan sistem yang telah dibangun ke dalam moodle sebagai fungsi tambahan.

## **B. Tinjauan Pustaka**

### **B.1 E-Learning**

Menurut (Som Naidu, 2006) menyatakan bahwa:

*“E-learning is commonly referred to the intentional use of networked information and communications technology in teaching and learning”*

E-learning sering disebut intensitas penggunaan informasi dan teknologi jaringan komunikasi dalam mengajar dan belajar. Sejumlah istilah lain juga digunakan untuk menggambarkan modus ini mengajar dan belajar. Mereka termasuk belajar online, belajar virtual, belajar didistribusikan, jaringan dan belajar webbased. Pada dasarnya, mereka semua mengacu pada proses pendidikan yang memanfaatkan teknologi informasi dan komunikasi untuk menengahi asinkron serta kegiatan belajar dan mengajar sinkron. Pada pengamatan lebih dekat, bagaimanapun, akan menjadi jelas bahwa label-label ini mengacu pada proses pendidikan yang sedikit berbeda dan karena itu mereka tidak dapat digunakan secara sinonim dengan istilah e-learning.

### **B.2 Pemeriksaan Otomatis**

Menilai tugas dan ujian berupa *source code* membutuhkan waktu dan tenaga yang besar. Banyak hal yang harus diperhatikan dalam pemeriksaan *source code* diantaranya adalah analisis leksikal dan analisis syntaks. Analisis leksikal cenderung menilai pada kecocokan tata bahasa sedangkan analisis syntaks adalah aturan penulisan *source code*. Dari hasil

pemeriksaan nantinya akan menghasilkan nilai yang diperoleh dari berapa jumlah yang benar dikurangkan dengan berapa jumlah error.

### B.3 Model Kompilator

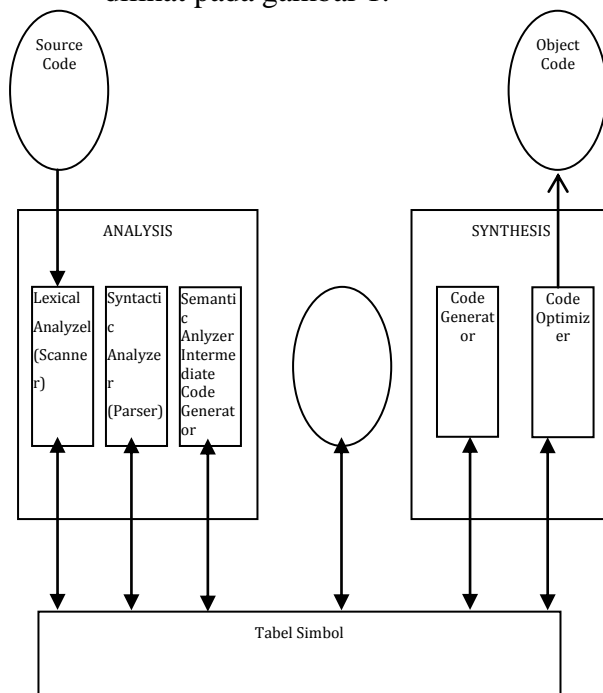
Pengembangan kompilator untuk sebuah bahasa merupakan pekerjaan yang kompleks (Farrar Utdirartatmo:2005). Hal tersebut dapat diatasi apabila perancang bahasa pemrograman mempertimbangkan bermacam-macam faktor perancangan. Sebuah kompilator memiliki dua tugas pokok yaitu sebagai berikut:

#### a. Fungsi Analisis

Fungsi ini bertugas melakukan dekomposisi program sumber menjadi bagian-bagian dasarnya atau biasa disebut sebagai Front End.

#### b. Fungsi Sintesis

Fungsi ini tugasnya adalah melakukan pembangkitan dan optimasi program objek atau biasa disebut sebagai *Back End*. Adapun model sebuah kompilator dapat dilihat pada gambar 1.



Gambar 1: Proses Kompilator

Keterangan Gambar

1. *Scanner*: memecah program sumber menjadi besaran leksikal/token.
2. *Parser*: memeriksa kebenaran dan urutan kemunculan token
3. *Analisis Semantik*: melakukan analisis semantik, biasanya dalam realisasi akan digabungkan dengan *intermediate code generator*.
4. *Code Generator*: membangkitkan kode objek
5. *Code Optimizer*: memperkecil hasil dan mempercepat proses
6. Tabel simbol: menyimpan semua informasi yang berhubungan dengan proses kompilasi.

## C. Metode penelitian

### C.1 Metode Analisis Data

Jenis penelitian yang dilakukan adalah bersifat eksperimental, yaitu membangun sistem pemeriksa *source code* otomatis bahasa pemrograman PHP. Sistem tersebut diuji dengan menggunakan jawaban tugas-tugas dan ujian mahasiswa/siswa yang menggunakan bahasa pemrograman PHP

### C.2 Metode Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah studi pustaka, untuk mendukung penyelesaian penelitian ini diperlukan beberapa referensi yang terkait dengan masalah yang akan diteliti. Angket dan wawancara, ditujukan untuk mengetahui apa dan bagaimana kebutuhan user dalam hal ini dosen/guru sebagai pengampu matakuliah khususnya matakuliah bahasa pemrograman PHP. Uji coba implementasi sistem, ditujukan tingkat kesesuaian antara sistem yang dibangun dengan kebutuhan user.

### C.3 Desain Penelitian

Sistem yang akan dibangun adalah sistem yang dapat memeriksa dan menilai *source code* bahasa pemrograman PHP dari segi strukturnya. *Source code* dinilai dalam 2 kategori penilaian yaitu secara *black box* dengan *white box*. Penilaian sintaks dilakukan dengan menggunakan *black box* dengan melalui tahap analisis leksikal dan analisis sintaks.

Sedangkan dalam perhitungan kompleksitas siklomatik dan jumlah *independent path*-nya digunakan *white box* dengan membentuk graf kontrol dari *source code*.

#### C.4 Teknik Pengujian

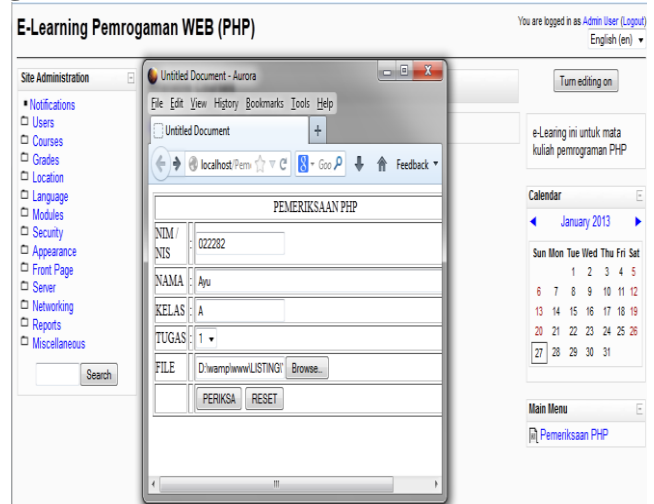
Untuk mengetahui keberhasilan suatu sistem maka dilakukan pengujian. Pengujian biasanya dilakukan dengan membandingkan sistem yang ada sebelumnya dengan sistem yang dibangun sekarang. Sistem dikatakan bagus apabila lebih menguntungkan jika dibandingkan dengan yang ada sebelumnya. Pengujian yang digunakan pada penelitian ini adalah pengujian dengan teknik *acceptance testing* berdasarkan kategori *functionality*. Pengujian ini ditujukan apakah sistem sudah sesuai dengan fungsinya. Pengujian dengan teknik *acceptance testing* berdasarkan kategori *performance*. Pengujian dengan teknik ini dimaksudkan untuk menguji kinerja suatu sistem misalnya keakuratan nilai jawaban mahasiswa.

### D. Hasil dan Pembahasan

#### D.1 Hasil Analisis Sistem Pemeriksaan Source Code

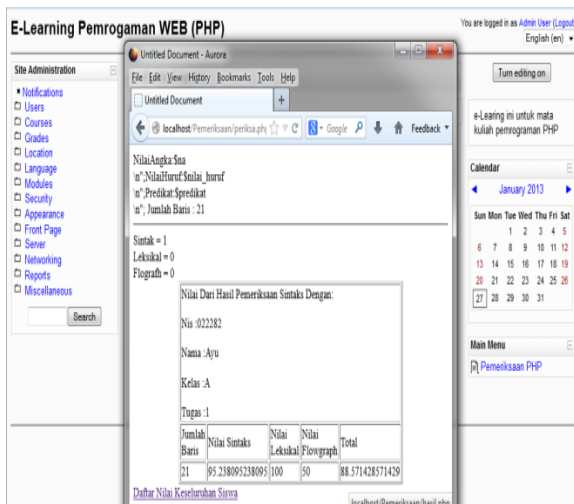
Berdasarkan desain sistem yang telah dibahas sebelumnya maka penulis dapat

merancang desain interface dan desain analisis sistem pemeriksaan *source code* bahasa PHP. Pertama-tama mahasiswa aktif *die-learning* kemudian mengklik link “Pemeriksaan” yang ada pada *e-learning* tersebut. Setelah Form isian sudah terbuka maka mahasiswa mengisi data dan menginput jawaban sesuai tugas yang dipilih seperti yang dapat dilihat pada **gambar 2**.



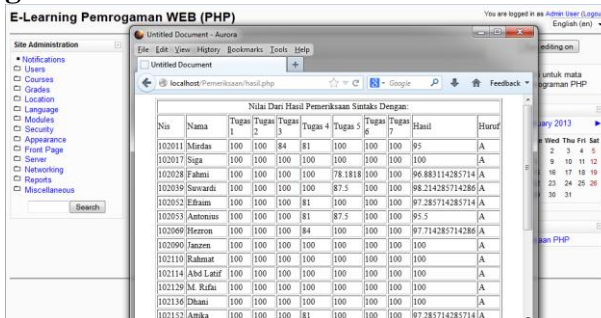
Gambar 2: Form Setelah Diisi dengan Data dan Jawaban Mahasiswa

Selanjutnya mahasiswa mengklik tombol “Periksa” dengan demikian jawaban dinyatakan masuk dalam tahap pemeriksaan di mana melalui 3 (tiga) tahap yaitu analisis leksikal, analisis sintaks dan analisis *flowgraph*. Setelah pemeriksaan dilakukan kemudian ditampilkan hasil pemeriksaan tersebut seperti pada **gambar3**.



Gambar 3: Hasil Penilaian Jawaban Mahasiswa

Mahasiswa dapat mengetahui kesalahan ada di mana dan nilai yang diperoleh. Selain nilai setiap tugas mahasiswa dan dosen pun dapat melihat rekapitulasi nilai seperti yang tampil pada gambar 4.



Gambar 4: Tampilan Rekapitulasi Nilai Untuk Semua Mahasiswa

## D.2 Pengujian Kategori functionality

Pengujian sistem menggunakan teknik acceptance testing untuk kategori functionality merupakan pengujian yang dilakukan berdasarkan fungsi-fungsi yang disediakan oleh sistem. Pada pengujian ini, dilakukan pengecekan apakah fungsi-fungsi tersebut telah memenuhi analisis identifikasi kebutuhan sistem yang telah dilakukan. Hasil pengujian sistem menggunakan teknik acceptance testing

untuk kategori functionality ditunjukkan pada Tabel 1 sebagai berikut:

Tabel 1: Hasil acceptance testing untuk kategori functionality

Identifikasi Kebutuhan	Terpenuhi	Keterangan
Input source code mahasiswa melalui Browser	√	- Penjelasan terdapat pada halaman ...
Dilakukan pengujian source code berdasarkan analisis leksikal	√	- Penjelasan terdapat pada halaman ... -
Dilakukan pengujian source code berdasarkan flowgraph	√	Penjelasan terdapat pada halaman
Output sistem berupa daftar hasil analisis	√	Penjelasan terdapat pada halaman

Hasil pengujian teknik acceptance testing untuk kategori functionality yang terdapat pada Tabel 1 menunjukkan bahwa sistem penilaian source code php telah memenuhi semua kebutuhan user sehingga tidak perlu dilakukan revisi terhadap sistem yang dibangun.

## D.3 Pengujian Kategori Performance

Pengujian sistem dengan kategori Performance merupakan pengujian melihat perbandingan hasil penilaian antara penilaian dosen secara manual dengan penilaian yang dihasilkan oleh sistem pemeriksa source code yang dibangun.

Contoh soal yang diuji dengan kategori performance adalah:

Buat sebuah program php dengan menggunakan pernyataan kondisi if...elseif dengan kasus Jika diketahui Nilai Angka dari nilai mahasiswa maka tampilkan "Nilai Huruf" dan "Predikat" yang diperoleh berdasarkan nilai angka dengan ketentuan jika nilai lebih besar 80 maka nilai huruf A, jika nilai antara 79 sampai 65 maka nilai huruf B, jika nilai antara 65 sampai 55 maka nilai huruf C, jika nilai antara 55 sampai 45 maka nilai

huruf D dan jika nilai kurang dari 45 maka nilai huruf E.

Hasil pengujian sistem menggunakan teknik acceptance testing untuk kategori performance ditunjukkan pada **tabel 2** sebagai berikut:

Tabel 2: Perbandingan hasil penilaian dosen dengan sistem untuk Tugas 4

Berdasarkan tabel 2 maka dapat disimpulkan bahwa terjadi selisih antara penilaian dosen dengan penilaian sistem pemeriksa otomatis.

#### D.4 Pembahasan

Penelitian ini menunjukkan bahwa pemeriksaan dilakukan dengan 3 tahap yaitu analisis leksikal, analisis sintaks dengan analisis flowgraph (Farrar Utdirartatmo, 2005). Penilaian untuk tahap analisis leksikal dilakukan berdasarkan jumlah error yang dihasilkan. Jika terdapat satu atau lebih kesalahan penulisan perintah (leksikal) dalam satu baris maka dianggap baris tersebut error. Penilaian error dihitung per baris dengan mengabaikan berapa jumlah error dalam baris tersebut.. Nilai tertinggi untuk tahap ini adalah seratus (100) dan nilai terendah adalah nol (0). Persamaan berikut menunjukkan cara mendapatkan nilai untuk tahap analisis leksikal.

$$\text{Nilai Leksikal} = 100 - \frac{\$kds}{\$jmlbrs} * (100 / (\$jmlbrs * 4))$$

Di mana:

*\$kds* adalah jumlah error leksikal.

*\$jmlbrs* adalah jumlah baris listing

Pada penilaian sintaks nilai tertinggi adalah 100, sedangkan nilai terendah adalah nol. Aturan penilaian pada tahap analisis sintaks sama yang diberlakukan pada tahap analisis leksikal. Persamaan berikut menunjukkan cara mendapatkan nilai untuk tahap analisis sintaks.

$$\text{Nilai Sintaks} = 100 - \frac{\$err}{\$jmlbrs} * (100 / \$jmlbrs)$$

Di mana:

*\$err* adalah jumlah error sintaks

*\$jmlbrs* adalah jumlah baris listing

Pada tahap analisis Flowgraph diberlakukan tiga (3) nilai yaitu nol (0), lima puluh (50) dan seratus (100). Jika nilai leksikal dan nilai sintaks adalah 100 maka nilai flowgraph juga 100. Jika nilai sintaks di tambah nilai leksikal

Nis	Hasil penilaian dosen	Hasil penilaian sistem	Selisih
102011	100	81	19
102017	100	100	0
102038	80	100	-20
102039	75	100	-25
102052	100	81	19
102053	100	81	19
102069	100	84	16
102090	90	100	-10
102110	100	100	0
102114	100	100	0

menghasilkan nilai lebih besar dari 50 maka nilai flowgraph adalah 50 sedangkan jika nilai sintaks dan nilai leksikal adalah 0 maka secara otomatis nilai flowgraph juga nol (0).

Nilai total dari hasil uji kebenaran source code diperoleh dengan persamaan sebagai berikut.

$$\text{Total} = 30\%(\text{Nilai Leksikal}) + 50\%(\text{Nilai Sintaksis}) + 20\%(\text{Nilai Flowgraph})$$

Persentase untuk nilai leksikal dan nilai sintaksis lebih besar dari yang diberikan untuk nilai flowgraph. Analisis leksikal dan analisis sintaks menguji kebenaran source code berdasarkan aturan bahasa pemrograman. Sedangkan analisis flowgraph lebih mengacu pada kebenaran logika source code dan dilakukan tanpa mempertimbangkan aturan bahasa pemrograman.

#### KESIMPULAN DAN SARAN

Sistem pemeriksaan *source code* bahasa pemrograman php yang dibangun seluruhnya berdasarkan analisis kebutuhan user dalam hal ini mahasiswa dan dosen. Pemeriksaan dilakukan dengan 3 (tiga)

tahap analisis yaitu analisis leksikal yang merupakan analisis terhadap ejaan keyword perintah, analisis sintaks yang menitikberatkan pada analisis kebenaran pola penggunaan listing dan analisis flowgraph atau logika yang menganalisa tentang kebenaran output. Proses pemeriksaan tersebut tersimpan dalam modul-modul pemeriksaan. Dari setiap tahap pemeriksaan setelah diuji dengan segi functionality dan performance maka diperoleh hasil bahwa sistem berjalan sesuai dengan procedure yang dibutuhkan oleh user dan sistem juga dapat diakses melalui *e-learning*.

Disarankan kepada peneliti selanjutnya agar mengembangkan sistem penilai source code PHP ini dengan mengikuti prosedur compiler yang lebih lengkap serta dapat mengintegrasikan dengan fitur-fitur tambahan dari *e-learning*.

## DAFTAR PUSTAKA

- Aleman, Jose Luise Fernandez.(2011). *Cross-Fertilization of Engineering Education research and Development of Automated Assesment in a Programming Tools Course, IEEE Transactions on education*, vol.54 No.4
- Naidu, Som. (2006). *E-Learning A Guidebook of Principles, Procedures and Practices*The University of Melbourne. Melbourne Victoria 3010 Australia
- Olivya, Meylanie. (2011). Sistem Penilai Source Code Otomatis Bersifat Generik. Makassar: Universitas Hasanuddin.
- Purbaningtyas, Rani. (2010). Pengembangan Prototype Sistem Penilaian Otomatis Mata Kuliah Pemrograman Java.Makassar. Universitas Hasanuddin.
- Qomaruddin, Muchammad Nur.(2009). Perancangan Sistem Visualisasi Automatis, Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya
- Tafiardi.(2005). Meningkatkan mutu pendidikan melalui e-learning, jurnal pendidikan penabur 04/Th.IV/Juni. Hal.85-97
- Utdirartatmo,Firrar.(2005). Teknik Kompilasi. Graha Ilmu. Yogyakarta.