

**IMPLEMENTASI PEMBELAJARAN ELEKTRONIK BAHASA  
PEMROGRAMAN PHP DENGAN PEMERIKSAAN SOURCE  
CODE OTOMATIS SEBAGAI PLUGIN MOODLE**

*Implementation of Electronic Learning of PHP Programming  
Language by Source Code Verification as Moodle Plugin*

**ANNAH**



**PROGRAM PASCA SARJANA  
JURUSAN TEKNIK ELEKTRO/INFORMATIKA  
UNIVERSITAS HASANUDDINMAKASSAR**

**2013**

**IMPLEMENTASI PEMBELAJARAN ELEKTRONIK BAHASA  
PEMROGRAMAN PHP DENGAN PEMERIKSAAN SOURCE  
CODE OTOMATIS SEBAGAI PLUGIN MOODLE**

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun dan diajukan oleh

Annah

Kepada

PROGRAM PASCASARJANA

UNIVERSITAS HASANUDDIN

MAKASSAR

2013

**TESIS**

**IMPLEMENTASI PEMBELAJARAN ELEKTRONIK BAHASA PEMROGRAMAN  
PHP DENGAN PEMERIKSAAN *SOURCE CODE* OTOMATIS SEBAGAI PLUGIN  
MOODLE**

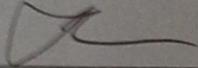
Disusun dan diajukan oleh

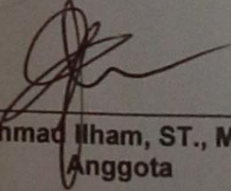
**ANNAH**

Nomor Pokok P2700210035

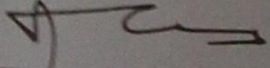
Telah dipertahankan di depan Panitia Ujian Tesis  
Pada tanggal 31 Januari 2013  
dan dinyatakan telah memenuhi syarat

Menyetujui  
Komisi Penasihat,

  
Dr. Ir. Zahir Zainuddin, MSc  
Ketua

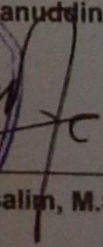
  
Amil Achmad Ilham, ST., MIT., Ph.D  
Anggota

Ketua Program Studi  
Teknik Elektro

  
Prof. Dr. Ir. H. Salama Manjang, MT

Direktur Program Pascasarjana  
Universitas Hasanudin



  
Mursalin, M.Sc

## **PERNYATAAN KEASLIAN**

Yang bertanda tangan dibawah ini:

Nama : Annah

Nomor Mahasiswa : P2700210035

Program Studi : Teknik Elektro

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut

Makassar, 15 Februari 2013

Yang menyatakan,

Annah

## KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT penulis haturkan atas segala karunia yang telah diberikan baik berupa kesehatan, kemudahan, maupun kemampuan sehingga penulis dapat menyelesaikan penyusunan hasil penelitian ini.

Tesis ini menawarkan solusi bagi pemeriksaan tugas siswa/mahasiswa khususnya yang berbasis bahasa pemrograman PHP yang mungkin dapat memberikan sedikit keringanan bagi dosen untuk memeriksa tugas mahasiswa utamanya untuk kelas yang jumlah mahasiswanya yang cukup besar sehingga dengan memasukkan *source code* maka dosen dapat langsung melihat hasil analisa tugas mahasiswa.

Selama penyusunan laporan ini dan pembuatan sistem penulisa banyak mengalami kendala namun berkat bantuan dan dukungan dari berbagai pihak maka penulis dapat menyelesaikannya. Ucapan terima kasih yang tak terhingga pula penulis haturkan kepada

1. Dr. Ir. Zahir Zainuddin, M.Sc, atas saran dan bimbingan teknis yang diberikan selama pengerjaan hasil penelitian ini.
2. Amil Ahmad Ilham, ST,MIT,P.hD, atas waktu yang diberikan untuk membimbing selama penyelesaian hasil penelitian ini.

3. Semua pihak yang telah membantu baik secara moril maupun material sehingga penulis merasa termotivasi dalam menyelesaikan penelitian ini.
4. Kepada teman-teman seperjuangan, kepada ibu, suami, anak dan keluarga tercinta terima kasih atas doanya.

Penulis menyadari bahwa penulisan hasil penelitian ini masih jauh dari kesempurnaan, masih terdapat kekurangan-kekurangan oleh sebab itu penulis sangat mengharapkan kritik dan saran konstruktif dari berbagai pihak. Semoga tulisan ini dapat bermanfaat bagi semua pihak.

Makassar,                      Februari 2013

ANNAH

## ABSTRAK

**ANNAH.** Implementasi Pembelajaran Elektronik Bahasa Pemrograman PHP dengan Pemeriksa Source Code Otomatis sebagai Plugin Moodle (dibimbing oleh Zahir Zainuddin dan Amil Ahmad Ilham).

Telah banyak lembaga pendidikan menerapkan sistem pembelajaran elektronik dengan menggunakan moodle untuk pengembangannya namun di moodle belum ada fitur untuk pemeriksaan listing program. Penelitian ini bertujuan untuk membangun suatu sistem yang dapat memeriksa tugas mahasiswa yang menggunakan bahasa pemrograman PHP dalam pembelajaran elektronik sehingga dapat lebih mengefisienkan waktu dosen dalam memeriksa tugas mahasiswa serta mahasiswa dapat langsung melihat nilainya pada saat itu juga sehingga transparansi nilai dapat dijamin dan sistem ini akan diintegrasikan ke moodle sebagai fitur tambahan.

Dalam penelitian ini digunakan metode eksperimental, dimana mahasiswa diberikan tugas melalui pembelajaran elektronik 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. Jika ditinjau dari segi waktu yang digunakan maka dapat disimpulkan bahwa pemeriksaan listing program dengan menggunakan sistem ini jauh lebih singkat dibandingkan dengan memeriksa secara manual dan keakuratan hasil pemeriksaan juga dapat lebih meningkat serta nilai mahasiswa juga dapat lebih transparan. Selain itu, pembelajaran elektronik yang menggunakan moodle juga dapat menggunakan sistem ini.

Kata kunci: Bahasa Pemrograman PHP, Pemeriksaan listing program

## ABSTRACT

**ANNAH**, *Implementation of Electronic Learning of PHP Programming Language by Source Code Verification as Moodle Plugin.* (supervised by Zahir Zainuddin and Amil Ahmad Ilham)

*Has many educational institutions to implement e-learning systems by using moodle moodle for development, but in no feature listings for examination program. This study aims to develop an application model as a tool to verify the correctness of the student programming tasks based on PHP programming language. Verification method using the tool in electronic learning can be very effitient for both lecturers in examining the studet works and students in receiving their task result, so that the grade of the tasks can be very clear to the students as their own efforts. The tool can also be integrated into application learning software, such as Moodle.*

*This study used an experimental method by giving students programming tasks through electronic learning. The students then send the answers in the form of source-code file into application program. After the answers' received, the application automatically verifies them and show results. Shipping answer task can only be done once. Experiments conducted on student assignment consisting of 20 students. System output is in the form of a number as grade level of students.*

*The results of this study indicate that the system can check the source-code of any application program based on PHP programming language. In the terms of the speed of the examination, it can be concluded that the verification to source-code using this system is much shorter than manually checking and accuracy of the results can also be increased besides that the value of student grades can also be more transparent. In addition, electronic learning using Moodle can also use this system.*

**Keywords :** *PHP Programming Language, Source Code Verification*



## DAFTAR ISI

HALAMAN JUDUL .....	i
KATA PENGANTAR .....	ii
ABSTRAK.....	iii
ABSTRACT .....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	viii
DAFTAR TABEL .....	ix
BAB I PENDAHULUAN .....	1
A. Latar Belakang Masalah .....	1
B. Rumusan Masalah.....	3
C. Tujuan Penelitian .....	4
D. Manfaat Penelitian .....	4
E. Batasan Masalah .....	5
F. Definisi Istilah .....	5
G. Organisasi/ Sistematika .....	6
BAB II TINJAUAN PUSTAKA .....	7
A. Pengertian Pembelajaran Elektronik.....	7
B. Moodle .....	9
C. Plugin .....	16
D. Automated Assessment in a Programming Tools Course.....	17

E. Mooshak .....	18
F. CourseMarker .....	19
G. BOOS.....	22
H. Pengembangan Prototype Sistem Penilaian Otomatis Matakuliah	
Pemrograman Java .....	25
I. Sistem Penilai Source Code Otomatis Bersifat Generik.....	26
J. Automated Assessment untuk Matakuliah Bahasa Pemrograman pada	
Sistem Pembelajaran Berbasis Web.....	27
K. Pemeriksaan Otomatis .....	28
L. Model Kompilator .....	28
<b>BAB III METODE PENELITIAN .....</b>	<b>34</b>
A. Rancangan Penelitian .....	34
B. Lokasi dan Waktu Penelitian .....	35
C. Bahan dan Alat Penelitian .....	35
D. Teknik Pengumpulan Data .....	36
E. Perancangan Sistem .....	36
F. Road Map Penelitian .....	36
G. Teknik Pengujian .....	39
H. Hasil Penelitian .....	39
<b>BAB III HASIL PENELITIAN DAN PEMBAHASAN .....</b>	<b>41</b>
A. Konsep kerja Sistem Pemeriksaan Source Code.....	41
B. Desain Sistem.....	42
C. Analisis Source Code .....	46
D. Implementasi Sistem Pemeriksaan Source Code .....	51
E. Hasil Pemeriksaan Source Code PHP.....	53

F. Pengujian Sistem Pemeriksaan Source Code PHP .....	57
BAB V KESIMPULAN DAN SARAN .....	66
A. Kesimpulan .....	66
B. Saran .....	66
DAFTAR PUSTAKA	
LAMPIRAN	

## DAFTAR GAMBAR

Gambar 1 Tampilan Awal Proses Instalasi Moodle .....	10
Gambar 2 Pengisian alamat web, tempat folder .....	11
Gambar 3 Pengisian nama database .....	11
Gambar 4 Moodle yang telah diinstal .....	12
Gambar 5 Matakuliah yang sudah diidi pada moodle .....	12
Gambar 6 Manajemen Power Suite .....	24
Gambar 7 Transmisi Management suite .....	25
Gambar 8 Manajemen Properti .....	25
Gambar 9 Model Kompilasi.....	29
Gambar 10 Kerangka konseptual .....	34
Gambar 11 Gambaran Umum Rancangan Sistem Penilai Source Code Otomatis .....	37
Gambar 12 Gambaran Umum Rancangan Sistem Pemeriksa Source Code Otomatis yang Dintegrasikan ke Moodle.....	37
Gambar 13 Konsep Kerja Sistem Pemeriksa Source Code .....	40
Gambar 14 Package Diagram Sistem Pemeriksaan Source Code .....	41
Gambar 15 Use Case Diagram Sistem Pemeriksaan Source Code .....	42

Gambar 16 Class Diagram Pemeriksaan Source Code Otomatis .....	43	
Gambar 17 Activity Diagram Sistem Pemeriksaan Source Code .....	44	
Gambar 18 Entity Relationship Sistem Pemeriksa Source Code .....	44	
Gambar 19 Contoh Pohon sintaks .....	47	
Gambar 20 Flowchart dan Flowgraph source code mahasiswa .....	48	
Gambar 21 Form Penginputan Data Mahasiswa dan Jawaban .....	51	
Gambar 22 Penginputan Data Mahasiswa dan Pengiriman Jawaban .....	52	
Gambar 23 Hasil Pemeriksaan source code Bahasa PHP untuk mahasiswa		54
Gambar 24 Rekapitulasi Hasil Pemeriksaan Jawaban Mahasiswa .....	55	
Gambar 25 Grafik Selisih Hasil Pemeriksaan secara manual dengan sistem...	59	
Gambar 26 Grafik waktu pemeriksaan secara manual .....	62	
Gambar 27 Grafik waktu pemeriksaan dengan sistem .....	63	
Gambar 28 Grafik efisiensi waktu hasil perbandingan antara pemeriksaan manual dengan sistem .....	65	

## DAFTAR TABEL

Tabel 1 Contoh Program Java .....	22
Tabel 2 Road Map Penelitian .....	38
Tabel 3 Contoh Source Code Mahasiswa .....	45
Tabel 4 Daftar Tipe Token .....	46
Tabel 5 Penentuan parameter flowgraf . .....	49
Tabel 6 Daftar perintah untuk uji flowgraph.....	51
Tabel 7 Hasil pengujian sistem untuk kategori functionality .....	57
Tabel 8 Perbandingan Hasil Penilaian dosen dan sistem untuk tugas 4.....	59
Tabel 9 Perbandingan waktu kerja manual dengan sistem .....	61

# BAB I

## PENDAHULUAN

### A. Latar Belakang Masalah

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.

Pengembangan pembelajaran elektronik terus dilakukan. Salah satu cara mudah untuk mengembangkan pembelajaran elektronik adalah digunakannya *moodle*. *Moodle* 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*.

Pada modul *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.

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 Lidemar Halide tahun 2010 mengembangkan penilaian otomatis untuk bahasa pemrograman berbasis web dengan menilai input-output dari tugas mahasiswa dan menghasilkan nilai berupa nilai huruf sehingga mahasiswa tidak dapat melihat nilainya secara detail. Rani Purbaningtyas tahun 2010 yang 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. Sedangkan peneliti berikutnya Meylanie Olivya tahun 2011 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 moodle 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.

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.

## **B. Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan sebelumnya maka yang menjadi rumusan masalah dalam penelitian ini adalah

1. Sistem pemeriksa *source code* yang telah dibangun oleh peneliti sebelumnya belum ada untuk pemeriksa *source code* bahasa pemrograman PHP.
2. Belum ada system yang terinclude dengan moodle yang dapat memeriksa *source code* PHP sehingga sistem e-learning yang menggunakan moodle belum dapat menggunakannya.

### **C. Tujuan Penelitian**

Adapun tujuan yang akan dicapai dalam penelitian ini adalah:

1. Untuk merancang sistem pemeriksa *source code* otomatis yang dapat memeriksa dan menilai program yang menggunakan bahasa pemrograman PHP.
2. Untuk mengintegrasikan sistem yang telah dibangun ke dalam moodle sebagai fungsi tambahan.

### **D. Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan manfaat terhadap:

1. Dosen/guru dapat menghemat waktu dalam memeriksa tugas-tugas/ujian mahasiswa/siswa serta dapat memudahkan dosen/guru untuk menyajikan materi dan soal khususnya yang berbasis bahasa pemrograman PHP.
2. Mahasiswa mempunyai kesempatan lebih banyak untuk menjawab soal yang berbasis pemrograman PHP.
3. Mahasiswa dapat melihat langsung nilai ujiannya sehingga keakuratan nilai mahasiswa dapat dijamin.
4. Bagi peneliti dapat mengembangkan penelitian ini di lain waktu

## **E. Ruang Lingkup/Batasan Penelitian**

Adapun batasan dalam penelitian ini mencakup:

1. Pembelajaran elektronik yang dimaksud adalah pembelajaran yang dilakukan melalui jaringan internet dengan menggunakan alat elektronik serta menyediakan layanan pembelajaran lainnya.
2. Sistem pemeriksa *source code* otomatis mencakup pemeriksaan yang dilakukan langsung oleh sistem dan menghasilkan suatu nilai.
3. Pemeriksa *source code* bahasa pemrograman PHP dibatasi pada analisis sintak, analisis leksikal, logika untuk tugas mahasiswa/siswa yang mengambil matakuliah Bahasa Pemrograman PHP.

## **F. Definisi dan Istilah**

Implementasi pembelajaran elektronik bahasa pemrograman PHP dengan pemeriksa *source code* otomatis sebagai plugin moodle merupakan sebuah modul yang dirancang untuk memeriksa *source code* dari tugas-tugas mahasiswa yang mengambil matakuliah bahasa pemrograman PHP yang nantinya akan diimplementasikan pada sistem pembelajaran elektronik dengan cara mengintegrasikan dengan moodle.

## **G. Organisasi/Sistematika**

### **BAB I      Pendahuluan**

Terdiri dari latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup/batasan penelitian, definisi dan istilah serta sistematika penulisan.

### **BAB II      Tinjauan Pustaka**

Terdiri dari tinjauan pustaka yang digunakan sebagai referensi secara teoritis untuk mendukung penelitian ini.

### **BAB III     Metode Penelitian**

Terdiri dari rancangan penelitian, waktu dan lokasi penelitian, bahan dan alat penelitian, teknik pengumpulan data, perancangan sistem, pengujian dan hasil penelitian.

### **BAB IV     Hasil Penelitian dan Pembahasan**

Bab ini berisi konsep kerja sistem pemeriksaan yang dibangun, desain sistem, pembuatan pemeriksaan source code PHP, implementasi system, dan pengujian system.

### **BAB V      Penutup**

Terdiri dari kesimpulan dan saran kegiatan penelitian yang telah dilaksanakan.

Daftar Pustaka

Lampiran

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Pengertian Pembelajaran Elektronik**

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.

Tafiardi (2005:87) mendefinisikan *e-learning* sebagai pembelajaran dengan menggunakan jasa bantuan perangkat elektronika. Fokus utama adalah proses belajarnya (*learning*) bukan pada “e” (*electronic*), karena perangkat elektronik hanya berperan sebagai alat bantu saja.

Berdasarkan pengertian di atas maka ada beberapa syarat suatu pembelajaran elektronik yaitu:

1. Kegiatan pembelajaran dilakukan melalui pemanfaatan jaringan ("jaringan" dalam uraian ini dibatasi pada penggunaan internet. Jaringan dapat saja mencakup LAN atau WAN). (Website eLearners.com).
2. Tersedianya dukungan layanan belajar yang dapat dimanfaatkan oleh peserta belajar, misalnya CD-ROM, atau bahan cetak.
3. Tersedianya dukungan layanan tutor yang dapat membantu peserta belajar apabila mengalami kesulitan.
4. Lembaga yang menyelenggarakan/mengelola kegiatan e-Learning.
5. Sikap positif dari peserta didik dan tenaga kependidikan terhadap teknologi komputer dan internet.
6. Rancangan sistem pembelajaran yang dapat dipelajari/diketahui oleh setiap peserta belajar.
7. Sistem evaluasi terhadap kemajuan atau perkembangan belajar peserta belajar. Mekanisme umpan balik yang dikembangkan oleh lembaga penyelenggara.

## B. Moodle

Melfachrozi M (2006) menjelaskan bahwa Moodle merupakan singkatan dari *Modular Object-Oriented Dynamic Learning Environment* yang berarti tempat belajar dinamis dengan menggunakan model berorientasi objek. Moodle merupakan salah satu aplikasi dari konsep dan mekanisme belajar mengajar yang memanfaatkan teknologi informasi, yang dikenal dengan konsep pembelajaran elektronik atau e-learning. Moodle dapat digunakan secara bebas sebagai produk sumber terbuka (open source) di bawah lisensi GNU. Moodle dapat diinstal di komputer dan sistem operasi apapun yang bisa menjalankan PHP dan mendukung database SQL.

Desain Moodle sangat sesuai untuk menjalankan kelas online yang mendukung pembelajaran tatap muka. Karakteristik lain dari desain Moodle adalah sebagai berikut (Muchammad Nur Qomaruddin:2009):

1. Sederhana, ringan, efisien, mudah diinstal, dan kompatibel.
2. Daftar kursus menampilkan deskripsi untuk setiap kursus yang tersimpan di server.
3. Sebagian besar area teks masukan (resources, forum postings, dan lain-lain) dapat diedit dengan editor WYSIWYG HTML.
4. Seluruh kursus dapat diklasifikasi dan dicari, satu situs Moodle dapat menyediakan ribuan kursus.
5. Menitikberatkan pada keamanan ketat secara keseluruhan. Setiap form dicek, data divalidasi, cookies dienkripsi, dan lainlain

## Langkah – langkah penginstalan :

1. Mengekstrak paket Moodle (berbentuk ZIP) yang telah didownload ke drive C:
2. Sebelum menginstall Moodle, harus mengaktifkan beberapa program yang telah ada satu paket dengan Moodle. Hal ini dilakukan agar Moodle bisa berjalan. Pertama sekali Aktifkan “setup\_xampp.bat”.
3. Jika instalasi xampp berakhir sukses, kemudian mulai aktifkan Apache 2 dengan mengklik (double klik) “apache\_start.bat”, Aktifkan juga MySQL dengan mengklik “mysql\_start.bat”. Gunakan mysql\_stop.bat untuk menonaktifkan MySQL server. Sedangkan Menonaktifkan Apache bisa dengan menutup jendela kerja Apache.
4. Mulai dengan membuka Internet Explorer dan ketiklah <http://127.0.0.1> atau <http://localhost> pada adress. Pada halaman pertama, terdapat pilhan bahasa yang diinginkan. Jika anda termasuk orang yang selalu sedih membaca bahasa Inggris karena susah mengartikannya anda dapat mengubahnya menjadi bahasa Indonesia. Dan akan ditampilkan halaman seperti berikut :

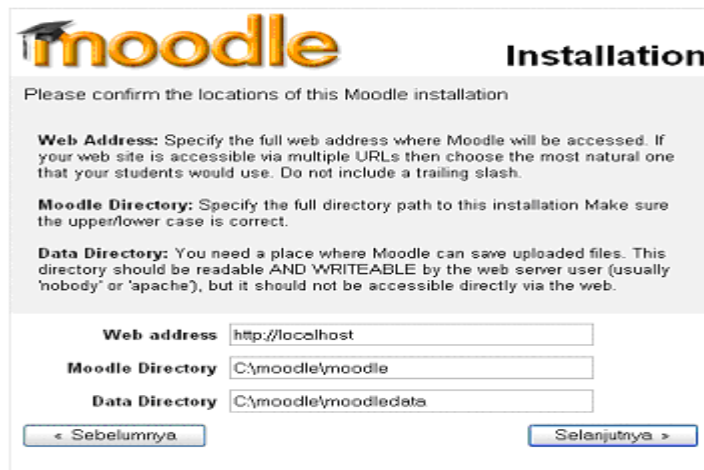


Gambar 1. Tampilan awal Proses Instalasi Moodle



Setelah Bahasa diubah menjadi Indonesia

5. Salah satu tahap dalam proses instalasi, akan disuruh untuk mengisi alamat web, tempat folder Moodle berada dan dimana tempat menyimpan data. Seperti tampak pada gambar.

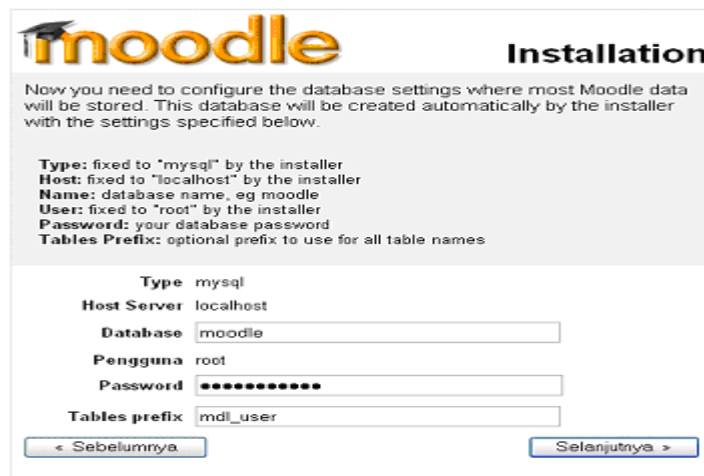


The screenshot shows the Moodle installation interface. At the top left is the Moodle logo, and at the top right is the word "Installation". Below the logo, there is a heading "Please confirm the locations of this Moodle installation". The main content area contains three sections of instructions: "Web Address" (specifying the full web address), "Moodle Directory" (specifying the full directory path), and "Data Directory" (specifying a place for uploaded files). Below these instructions are three input fields: "Web address" with the value "http://localhost", "Moodle Directory" with the value "C:\moodle\moodle", and "Data Directory" with the value "C:\moodle\moodledata". At the bottom, there are two buttons: "< Sebelumnya" and "Selanjutnya >".

Gambar 2. Pengisian alamat web, tempat Folder

Moodle berada dan tempat menyimpan data

6. Anda tinggal memberi password yang mudah anda ingat, karena nama database yang harus diisi, sudah terisi dengan sendirinya.



The screenshot shows the Moodle installation interface for database configuration. At the top left is the Moodle logo, and at the top right is the word "Installation". Below the logo, there is a heading "Now you need to configure the database settings where most Moodle data will be stored. This database will be created automatically by the installer with the settings specified below." The main content area contains a list of settings: "Type: fixed to 'mysql' by the installer", "Host: fixed to 'localhost' by the installer", "Name: database name, eg moodle", "User: fixed to 'root' by the installer", "Password: your database password", and "Tables Prefix: optional prefix to use for all table names". Below these settings are five input fields: "Type" (mysql), "Host Server" (localhost), "Database" (moodle), "Pengguna" (root), and "Password" (masked with dots). The "Tables prefix" field contains "mdl\_user". At the bottom, there are two buttons: "< Sebelumnya" and "Selanjutnya >".

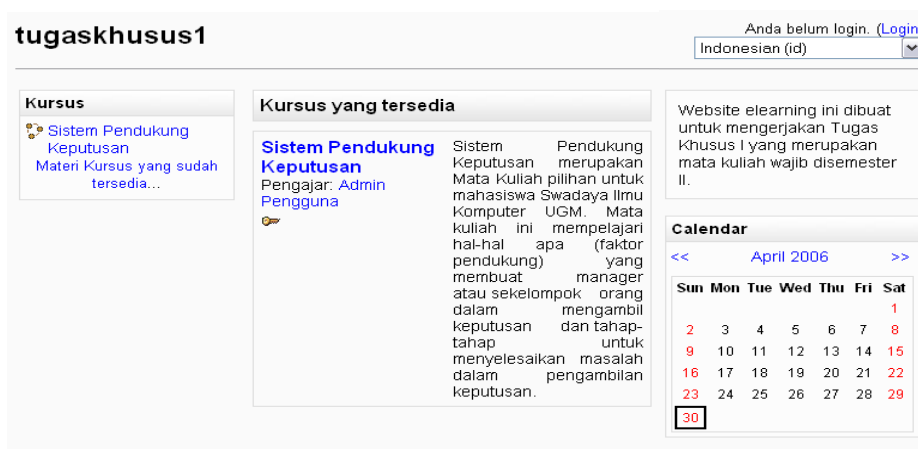
Gambar 3. Pengisian nama database dan password

7. Setelah proses instalasi selesai, maka tampilan web akan seperti gambar 4. Tampak Admin sebagai pengguna yang aktif.



Gambar 4. Moodle yang telah diinstall

8. Selanjutnya kita tinggal menambah pelajaran yang kita inginkan, membuat bahan yang akan diajar, membuat soal – soal untuk kuis, dan hal lain sebagainya. Kita tinggal menambah atau mengurangi apa yang telah ada.
9. Sistem Pendukung Keputusan merupakan mata kuliah yang dibuat sebagai contoh dalam menggunakan Moodle. Tampak gambar sedikit berbeda dengan gambar 4, hal ini dikarenakan belum ada satupun pengguna yang sedang aktif untuk mengikuti kegiatan belajar perkuliahan, termasuk juga administrator.



Gambar 5. Mata Kuliah yang sudah diisi pada Moodle

## **Management moodle**

### **1. Site Management**

- a. Web site diatur oleh admin, yang telah ditetapkan ketika membuat website.
- b. Tampilan (themes) diizinkan pada admin untuk memilih warna, jenis huruf, susunan dan lain sebagainya untuk kebutuhan tampilan.
- c. Bentuk kegiatan yang ada dapat ditambah.
- d. Source Code yang digunakan ditulis dengan menggunakan PHP. Mudah untuk dimodifikasi dan sesuai dengan kebutuhan.

### **2. User management**

- a. Ini digunakan untuk mengurangi keterlibatan admin menjadi lebih minimum, ketika menjaga keamanan yang berisiko tinggi.
- b. Metode email standar: di mana, pelajar dapat membuat nama pemakai untuk login. Alamat email akan diperiksa melalui konfirmasi.
- c. Tiap orang disarankan cukup (1) pengguna saja untuk seluruh sever. Dan tiap pengguna dapat mempunyai akses yang berbeda.
- d. Pengajar mempunyai hak istimewa, sehingga dapat mengubah (memodifikasi) bahan pelajaran.
- e. Ada "kunci pendaftaran" untuk menjaga akses masuk dari orang

yang tidak dikenal.

- f. Semua Pengguna dapat membuat biografi sendiri, serta menambahkan photo.
- g. Setiap pengguna dapat memilih bahasa yang digunakan.

### **3. Course management**

- a. Pengajar mempunyai kendali secara penuh untuk mengatur pelajaran, termasuk melarang pengajar yang lain.
- b. Memilih bentuk/metode pelajaran seperti; berdasarkan mingguan, topik atau bentuk diskusi.
- c. Terdapat Forum, Kuis, Polling, Survey, Tugas, Percakapan dan Pelatihan yang digunakan untuk mendukung proses belajar.
- d. Semua kelas-kelas untuk Forum, Kuis dan Tugas dapat ditampilkan pada satu halaman (dapat didownload sebagai file lembar kerja).
- e. Bahan pelajaran dapat dipaketkan dengan menggunakan file zip.

### **Pengembangan moodle**

Menurut (Muchammad Nur Qomaruddin:2009) Pengembangan Moodle dibagi dalam tiga macam kontribusi yaitu membuat sebuah plugin baru, mengubah kode inti, dan beberapa kontribusi yang tidak melibatkan pemrograman PHP.

## 1. Membuat plugin baru

Huruf M di Moodle mewakili kata modular dan kemudahan. Cara yang paling mudah dipelajari untuk menambah fungsi baru di Moodle adalah dengan menggunakan salah satu dari *plugin API*.

## 2. Mengubah kode inti Moodle

Beberapa tipe perubahan hanya dapat dilakukan dengan mengedit kode inti Moodle. Perubahan ini jauh lebih sulit dilakukan daripada penambahan plugin. Perubahan kode inti Moodle merupakan salah satu kegiatan pengembangan utama yang sebaiknya dilakukan dengan melakukan beberapa tahap berikut.

- a. Memastikan bahwa ide yang diusulkan adalah patut dilakukan dengan mendiskusikan ide tersebut dengan para pengembang Moodle melalui forum atau media yang lain.
- b. Membuat satu spesifikasi di Moodle Docs yang memuat semua Informasi atau dokumentasi kegiatan pengembangan yang dilakukan secara detail.
- c. Memperhatikan dan memahami umpan balik dari segenap komunitas Moodle terhadap pengkodean yang dilakukan.
- d. Mendaftar tugas-tugas di Moodle Tracker yang merepresentasikan progres kegiatan pengembangan Moodle agar dapat diamati oleh segenap komunitas Moodle.
- e. Meminta para tester (penguji) untuk mengamati, mencoba, dan mencari kekurangan dari kode yang sedang dikembangkan.

- f. Merespon segala laporan yang menyatakan bug (kesalahan) yang ditemukan di kode yang sedang dikembangkan.
3. Kontribusi yang tidak melibatkan pemrograman PHP  
Hal ini dapat dilakukan dengan tahap:
    - a. Membuat tema Moodle
    - b. Menerjemahkan Moodle ke bahasa yang lain
    - c. Membantu dokumentasi Moodle
    - d. Membantu proses pengujian

### **C. Plugin**

Menurut Codex, plugin adalah seperangkat alat yang digunakan untuk menambah/memperluas kemampuan WordPress. Berikut keuntungan penggunaan/pengembangan plugin:

1. Tidak memodifikasi *Core* aplikasi. Mengubah sebuah skrip di *core* aplikasi yang sering terupdate sangat tidak efektif. Belum lagi jika apa yang anda ubah itu tak terdokumentasi dan tak diketahui kembali dimana dan kenapa diubah.
2. *Don't Reinvent the Wheel*. Dengan plugin, kita tak perlu menulis kembali fungsi-fungsi yang sudah cukup banyak ditawarkan WordPress.
3. Pemisahan antara tampilan (themes) dan plugin. WordPress memisahkan dari awal apa itu themes dan plugin sehingga

pengembang tidak menaruh banyak fungsi dan skrip-skrip spesifiknya di themes atau ditemplate.

4. Mudah diupdate. Jika plugin anda sudah masuk di repository maka pengguna plugin kita akan lebih mudah mengupdate plugin yang kita buat. Jikapun tidak berada di repository, WordPress menyediakan fasilitas update plugin dengan mengupload (cukup) sebuah file Zip berisi plugin yang dimaksud.
5. Mudah digunakan dan dibagi.
6. Sandbox. Pengembang plugin tak perlu khawatir akan plugin yang dikembangkan bisa merusak konfigurasi dan content dari situs yang dikelolanya. WordPress menyediakan enviroment sandbox yang bagus, cukup load file *wp-load.php* dan anda sudah mendapati core WP secara keseluruhan.
7. Komunitas Pengembang yang luas

#### **D. *Automated Assessment in a Programming Tools Course***

Hasil penelitian ini menyatakan bahwa sistem penilaian otomatis dapat berguna untuk para siswa dan instruktur. Umpan balik meningkat dan dapat dengan cepat memiliki efek yang positif terhadap pembelajaran siswa. Tulisan ini menyajikan sebuah pengalaman menggunakan penilaian otomatis dalam kursus tools pemrograman. Penelitian ini bertujuan untuk memperluas penggunaan sistem penjurian online yang tradisional dengan serangkaian tugas yang berhubungan dengan tools pemrograman. Beberapa hasil empiris tentang bagaimana siswa

menggunakan sistem penilaian secara otomatis dalam kursus CS2 disajikan. Penelitian menunjukkan bahwa sistem penilaian secara otomatis memperkenalkan minat siswa dan menghasilkan perbedaan statistik yang signifikan dalam skor antara eksperimen dan grup kontrol.

Pendekatan penilaian secara otomatis diterapkan pada kursus tools pemrograman telah disajikan. Dalam studi terkontrol, nilai dari grup eksperimen menunjukkan bahwa siswa memperoleh graps lebih tepat dari konsep yang berkaitan dengan *debugging*, penyebaran, dan *versioning*. Ini jelas mendukung gagasan bahwa penilaian otomatis mempromosikan kemampuan siswa juga untuk belajar tool pemrograman.

### **E. *Mooshak***

*Mooshak* adalah sistem untuk mengelola kontes pemrograman di Web. Fitur dasar *Mooshak* meliputi penilaian otomatis terhadap program yang disampaikan, menjawab pertanyaan yang merupakan klarifikasi tentang deskripsi masalah, mengevaluasi ulang program, melakukan pencetakan pelacakan kesalahan, dan lain-lain. *Mooshak* mendukung berbagai jenis kontes, khususnya yang didasarkan pada aturan ICPC. Hal ini juga mendukung untuk IOI dan dapat dengan mudah diperluas untuk jenis kontes yang baru.

Sistem ini pada awalnya ditujukan untuk kontes, tetapi semakin banyak digunakan dalam kursus pemrograman untuk memberikan umpan balik secara langsung terhadap tugas, untuk menerima dan memvalidasi tugas, untuk pra-evaluasi dan memeriksa tugas, dan lain-lain



## F. CourseMarker

*CourseMarker* adalah sistem yang dikembangkan di Sekolah Ilmu Komputer dan Teknologi Informasi untuk penilaian otomatis tugas pemrograman (Foxley et al., 2001, Higgins dkk., 2002a). Hal ini didasarkan pada sistem terdahulu, Ceilidh, tetapi merupakan implementasi ulang yang lebih lengkap. Selain penilaian pemrograman (terutama dalam bahasa Java) *CourseMarker* baru-baru ini telah diperpanjang untuk memeriksa tugas diagramatic (Higgins dkk., 2002b).

*CourseMarker* (Ceilidh) berbeda dengan kebanyakan paket CAA lain karena mereka menilai program-program yang berupa tugas-tugas bukan jawaban atas pertanyaan. Hal ini disorot oleh komentar pada *CourseMarker* dalam survei terbaru dari metode CAA.

*CourseMarker* adalah sistem untuk kursus berbasis client-server untuk pemrograman atau diagram latihan. Ini memberikan solusi untuk memberikan spesifikasi pembelajaran, memungkinkan siswa untuk mengembangkan program secara bebas dan otomatis menandai dan memantau respon dari siswa. Tidak seperti perangkat lunak lain *CourseMarker* disesuaikan dengan kebutuhan program pengajaran, dan telah dikembangkan dan digunakan selama beberapa tahun.

*CourseMarker* merupakan versi update dari server yang telah diuji (10 tahun) atau sistem klien berbasis Ceilidh. Ia memiliki sejumlah kemampuan yang unik, misalnya penilaian diagram berbasis kerja. Ini adalah akurat dan mudah digunakan, terutama untuk siswa. Dosen diberikan berbagai macam statistik, meskipun melalui antarmuka yang sedikit rumit. Dapat menandai secara

fleksibel, meskipun dibangun pada tools yang inovatif. Instalasi mungkin menjadi masalah internal di mana tergantung pada platform dan konfigurasi yang digunakan di setiap institusi. Semua aspek penggunaannya sepertinya telah bagus, dan keamanan sudah dijamin.

Adapun perangkat CourseMarker adalah sebagai berikut:

1. Perangkat tipografi: Ini menganalisis pengajuan untuk layout termasuk indentasi dan penggunaan lainnya dari spasi, komentar, dan penggunaan variabel. Tata letak dari sebuah program dinilai karena indicitive gaya pemrograman.
2. Tool yang Dinamis: Ini menganalisis tugas dengan menjalankan program pada set data uji. Ini menentukan apakah tugas benar-benar jalan.
3. Perangkat Fitur: Fitur memeriksa apakah konstruksi pemrograman tertentu telah digunakan. Hal ini berguna karena beberapa latihan yang dirancang untuk menggambarkan konstruksi tertentu, seperti looping. Selain itu ada tools penilai lainnya untuk tugas non-program.

Tiga alat pemeriksa yang digunakan untuk membangun sebuah skema penilaian untuk sebuah tugas. Tools yang berbeda, dan aspek yang berbeda dari setiap tools, dapat dikombinasikan untuk menghasilkan nilai akhir. Tools ini juga dapat untuk periksa scaling, di mana diperlukan, agar sesuai dengan distribusi yang diharapkan. Selain itu, *CourseMarker* dapat dikonfigurasi untuk menerima kiriman ganda. Hal ini memungkinkan siswa untuk mencoba mengirim jawaban

tugas kemudian menunggu umpan balik. Setelah ada umpan balik maka siswa dapat mengirim jawaban kembali. Untuk penilaian sumatif jumlah pengiriman yang diizinkan untuk setiap siswa biasanya akan terbatas, untuk mencegah coba-kesalahan pemrograman, tetapi hal ini dapat flexibel untuk penilaian formatif.

Sistem juga disediakan untuk memberikan umpan balik kepada siswa atas dasar tugas mereka. Pengalaman dengan *CourseMarker* menunjukkan bahwa mahasiswa lebih suka sistem untuk menentukan kesalahan yang tepat dari jawaban tugas mereka, tapi "umpan balik yang terlalu banyak dapat merusak proses belajar siswa" (Higgins dkk., 2002a). Berbagai bentuk umpan balik dapat diberikan, termasuk saran untuk perbaikan jawaban secara langsung dan menunjuk ke materi pembelajaran yang relevan.

*CourseMarker* menggunakan metode cukup mudah untuk mendeteksi plagiarisme. Tugas yang dikirim secara berkelompok besar kemungkinan bisa sama. Ini dapat mengambil beberapa waktu untuk memeriksa kesamaan dalam kelas yang besar, jika ada mahasiswa  $n$  maka ada  $(n(n-1)) / 2$  yang perlu diperiksa. Deteksi plagiarisme membandingkan pekerjaan siswa dalam hal kesamaan. Proses ini juga memperhitungkan fakta bahwa beberapa aspek program, seperti nama variabel, komentar, dan spasi, dapat diubah tanpa mengubah program dengan cara yang nyata. Sebagai contoh, berikut dua buah kode Java,

walaupun terlihat sangat berbeda, yang persis sama adalah compiler yang bersangkutan:

Tabel 1: Contoh program Java

<pre>public void printHello() {     // Prints Hello World to     standard out String     str = "Hello World";     // A string to print     System.out.println(str);     // Print it }</pre>	<pre>public void sayHi() {     /* Doesn't look like the other one     */ String myString="Hello World";     System.out.println(myString); }</pre>
---	---

*CourseMarker* membandingkan program dari tugas siswa dan memberikan laporan seberapa mirip dari masing-masing tugas, dan apa perbedaan antara tugas mereka. Tingkat perbedaan menunjukkan seperti apa perubahan (menghapus spasi, mengubah nama variabel, dll) perlu dilakukan untuk membuat program yang sama. Hasilnya berdasarkan kesamaan sehingga tugas yang paling mirip merupakan fokus pemeriksaan. Sebuah ekstensi baru untuk *CourseMarker* adalah kemampuan untuk menilai courseworks diagramatic (Higgins dkk., 2002b).

## G. BOOS

BOSS (Broadband Operasi Software Solutions) adalah sistem perangkat lunak modular yang menyediakan solusi otomatis untuk operasi jaringan HFC. Kebanyakan media untuk perusahaan besar jaringan HFC memahami bahwa mereka membutuhkan komputerisasi pusat operasi lokal, regional, dan nasional yang akan memberikan peringatan dini masalah jaringan dan memungkinkan pemeliharaan just-in-time.

Sayangnya, solusi perangkat lunak manajemen jaringan saat ini tersedia untuk operator jaringan memiliki akar dalam telekomunikasi atau industri jaringan data, dan tidak sepenuhnya sesuai untuk mengelola jaringan HFC. BOSS adalah sistem prediktif dan proaktif yang menggunakan informasi dari beragam koleksi sistem perangkat lunak yang ada di seluruh perusahaan, dan menawarkan solusi untuk masalah jaringan operasi. BOSS menyediakan serangkaian platform untuk menarik operasi ini bersama-sama, baik secara efektif dan efisien. Desain modular memungkinkan operator untuk membeli hanya modul individu yang dibutuhkan saat ini, sedangkan memungkinkan untuk ekspansi di masa depan.

Sebagai jaringan berkembang, modul perangkat lunak lain yang ditambahkan yang diperlukan untuk menyediakan integrasi data dan fungsionalitas tambahan. BOSS merupakan 'perubahan paradigma' yang benar untuk menyediakan solusi perangkat lunak otomatisasi untuk perusahaan jaringan. BOSS sangat fleksibel dan desain modular memungkinkan operator untuk hanya membeli modul solusi individu yang menambah nilai pada operasi mereka untuk kebutuhan saat ini, dengan fleksibilitas untuk tumbuh dan berubah sebagai kebutuhan mereka untuk berevolusi. Solusi BOSS masing-masing ditargetkan pada spesifik tantangan operasi jaringan dunia nyata, memberikan perbaikan nyata dalam efisiensi tenaga kerja, ketersediaan jaringan, dan return-on-

investasi.

Di antara produk BOSS, yang dirilis saat ini termasuk komponen seperti:

a. Manajemen Power Suite

Suite Power Management adalah sistem manajemen daya lengkap, meliputi aspek jaringan listrik, baterai siaga, dan pasokan listrik. Ini menyediakan pelaporan real-time, kontrol proaktif, dan catatan sejarah lengkap dari semua data yang terkait dengan daya.



Gambar 6: Manajemen Power Suite

b. Transmisi Management Suite

Suite Manajemen Transmisi menyediakan alat untuk menjamin kelangsungan dan kualitas sinyal serta untuk secara proaktif mengurangi masukan. Modul seperti Pemanfaatan Manager Bandwidth dan Manajer Patahan mempertahankan pemanfaatan maju dan mundur dari berbagai node dalam suatu sistem.



Gambar 7: Transmisi Management Suite

- c. Manajemen Properti dan komponen Manajemen HFC juga sedang dikembangkan. Informasi lebih lanjut akan diposting setelah tersedia



Gambar 8: Manajemen Properti

## H. Pengembangan Prototype Sistem Penilaian Otomatis Matakuliah Pemrograman Java

Penelitian yang dilakukan oleh Rani Purbaningtyas (2010) ini bertujuan mengembangkan prototype awal sistem penilaian otomatis matakuliah bahasa pemrograman Java yang digunakan untuk memeriksa tugas bahasa pemrograman Java mahasiswa. Prototype yang dibangun dijadikan dasar untuk pengembangan sistem penilaian otomatis matakuliah pemrograman Java yang sempurna.

Secara *white box checking*, sistem akan melalui tahap pemeriksaan eksekusi program, tahap pemeriksaan logika program dengan cara

melakukan pengujian input-output, dan tahap pemeriksaan *source code*. Metodologi yang diterapkan untuk pemeriksaan *source code* tugas bahasa pemrograman Java mahasiswa menggunakan teknik pencocokan string (*string matching*) yaitu *regular expression*.

Secara *black box checking* hanya akan melalui 2 tahap yaitu tahap pemeriksaan eksekusi program dan tahap pemeriksaan dan tahap pemeriksaan logika program dengan cara melakukan pengujian input-output. Keluaran sistem berupa nilai angka yang menunjukkan tingkat kesempurnaan modul program yang dibuat mahasiswa akan disimpan dalam file tersendiri berekstensi "txt".

Adapun yang belum dicapai oleh penelitian Rani Purbaningtyas adalah masih perlu dikembangkan referensi yang dapat mengakomodasi pemberian nilai input terhadap program mahasiswa sehingga mahasiswa tidak perlu mempelajari operasi file terlebih dahulu untuk pengerjaan tugas pembuatan program. Inputan belum berupa *flowchart* program atau *pseudocode* agar dapat menjadi variasi bagi dosen pada saat pemberian tugas sehingga dosen tidak perlu membuat kunci jawaban tugas.

## **I. Sistem Penilai Source Code Otomatis Bersifat Generik**

Penelitian yang dilakukan oleh Meylanie Olivya (2011) ini bertujuan membangun sistem penilai otomatis yang dapat menganalisis dan menilai *source code* dalam berbagai bahasa pemrograman dan mengimplementasikan perangkat lunak yang telah dibangun agar dapat



melakukan penilaian terhadap tugas-tugas dan ujian matakuliah yang berbasis pemrograman.

Sistem yang dibangun memeriksa *source code* mahasiswa melalui tiga tahap pengujian kebenaran *source code* yaitu tahap analisis leksikal, tahap analisis flowgraph, dan tahap analisis sintaks. Dengan melalui tiga tahap ini, diharapkan bahwa *source code* yang dianalisis dapat ditentukan kebenarannya meskipun tanpa dilakukan kompilasi.

Adapun hal yang belum tercapai dalam penelitian ini adalah sistem belum merupakan sistem kompiler yang utuh dan belum dapat memeriksa bahasa pemrograman yang lain seperti PHP.

#### **J. Automatic Assessment untuk Mata Kuliah Bahasa Pemrograman Pada Sistem Pembelajaran Berbasis Web**

Penelitian ini dikembangkan oleh Lidemar Halide(2010) menunjukkan hasil system Automatic Assessment (AA) yang dirancang dengan menggunakan bahasa pemrograman PHP dengan 3 model yaitu AA langsung menampilkan hasil, diberikan 1 macam input dan 2 macam input yang berbeda (angka dan huruf). Cara kerja dari system ini adalah user menginput nilai kemudian system langsung menampilkan hasilnya dan system memberi nilai berupa nilai huruf dan salah atau benar. Kekurangan dari system ini mahasiswa tidak dapat melihat nilai angkanya jadi tidak ada persentasi nilai. Jika ada kesalahan jawaban maka mahasiswa tidak memperoleh nilai.

## **K. 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 sintaks. Analisis leksikal cenderung menilai pada kecocokan tata bahasa sedangkan analisis sintaks adalah aturan penulisan *source code*. Dari hasil pemeriksaan nantinya akan menghasilkan nilai yang diperoleh dari berapa jumlah yang benar dikurangkan dengan berapa jumlah error.

## **L. Model Kompilator**

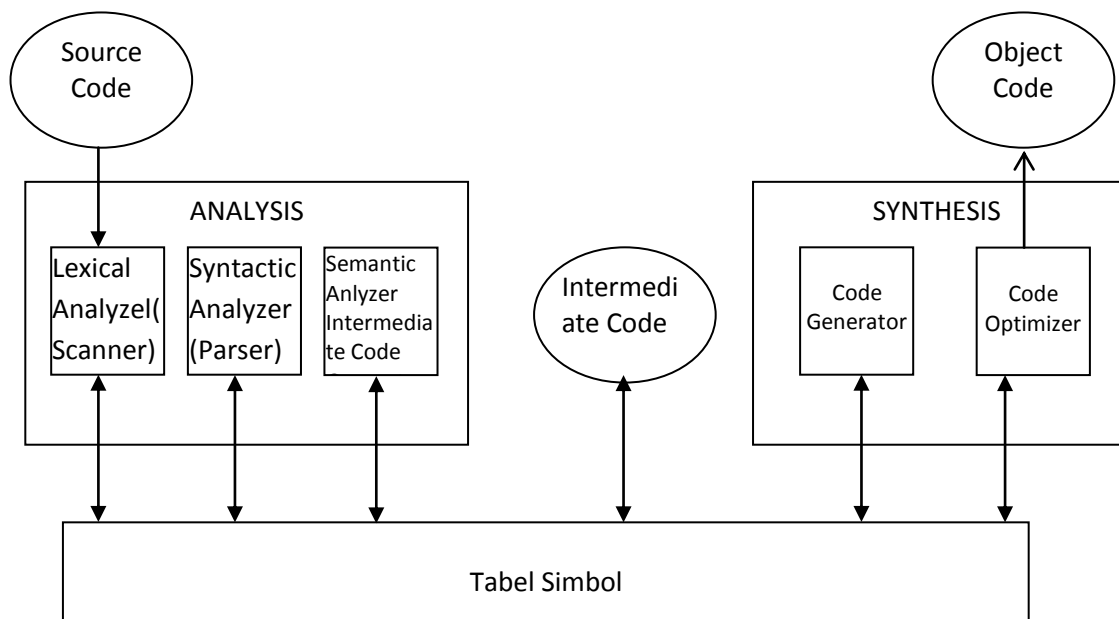
Pengembangan kompilator untuk sebuah bahasa merupakan pekerjaan yang kompleks (Firrar Utdirartatmo:2005). Hal tersebut dapat diatasi apabila perancang bahasa pemrograman mempertimbangkan bermacam-macam faktor perancangan. Sebuah kompilator memiliki dua tugas pokok yaitu sebagai berikut:

### **1. Fungsi Analisis**

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

## 2. Fungsi Sintesis

Fungsi ini tugasnya adalah melakukan pembangkitan dan optimasi program objek atau biasa disebut sebagai *Back End*. Adapun model sebuah kompiler dapat dilihat pada gambar 8 sebagai berikut:



Gambar 9: Model Kompiler

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.

## 1. Analisis Leksikal

Menurut Firrar Utdirartatmo(2005:43) menyatakan bahwa analisis leksikal atau scanner bertugas mengidentifikasi semua besaran pembangun bahasa (leksikal) yang ada pada kode sumber (*source code*). Scanner menerima masukan kode sumber berupa serangkaian karakter kemudian memilah-milahnya ke dalam satuan leksikal, yaitu token.

Token-token ini akan menjadi masukan bagi analisis selanjutnya yaitu analisis sintaksis.

Dari fungsi scanner secara umum seperti telah disebutkan di atas, maka tugas scanner secara rinci adalah:

- a. Membaca serangkaian karakter dari kode sumber
- b. Mengenalinya ke dalam satuan leksikal
- c. Mengubahnya menjadi token dan menentukan jenis tokennya.
- d. Mengirimkan token ke proses analisis selanjutnya, yaitu analisis sintaksis
- e. Mengabaikan karakter *white space* (spasi, enter, ganti baris, penanda akhir file) dan komentar (*remark*) apabila ada di dalam kode sumber
- f. Menangani kesalahan
- g. Menangani tabel simbol

Besaran pembangun bahasa (leksikal) meliputi:

- a. Identifier

Identifier atau pengenalan dapat berupa:

- 1) Kata tercadang atau kata kunci (*reserve word*) yang telah didefinisikan oleh bahasa pemrograman
- 2) Kata yang dideklarasikan sendiri oleh pemakai (si pembuat program) disebut dengan variabel.

Contoh (dalam bahasa Pascal)

```
var
  nama : string
  alamat : string
  jumlah_anak : byte
  nominal_tabungan : real
begin
  clrscr;
end.
```

Dari contoh di atas, maka identifier yang dideklarasikan sendiri oleh pemakai (variabel) adalah: nama, alamat, jumlah\_anak, dan nominal\_tabungan. Sedangkan identifier kata tercadang (*reserve word*) adalah: var, string, byte, real, begin, clrscr, dan end. Selain contoh yang disebutkan di atas, masih banyak kata tercadang yang lainnya.

Untuk mendeklarasikan identifier oleh pemakai setiap bahasa pemrograman memiliki aturan yang berbeda-beda, demikian juga dengan identifier kata tercadang memiliki nama dan arti yang berbeda untuk setiap bahasa pemrograman.

Contoh

begin dan end

di bahasa Pascal → kata tercadang

di bahasa C → boleh digunakan untuk nama variabel

{ dan }

di bahasa Pascal → 'kata' lebih tepatnya delimiter atau pembatas untuk memberikan komentar pada kode sumber.

di bahasa C → delimiter untuk mengawali dan mengakhiri sebuah blok program

#### b. Nilai konstanta

Nilai konstanta disini dapat berupa: integer, real, boolean, character, string, dan sebagainya. Melalui contoh berikut ini, bedakan antara nilai (atau isi) dengan variabel yang digunakan untuk menyimpannya.

Contoh (dalam bahasa Pascal)

```
hitung_luas := panjang * lebar;  
x := 1000;  
ulang := false;  
luas_lingkaran := 3.14 * jejari * jejari;  
nama:= budi hartono;
```

Dari contoh di atas, maka: 1000, false, 3.14, 'budi hartono' termasuk besaran leksikal berupa nilai konstanta. Sedangkan hirung\_luas, panjang, lebar, x, ulang, luas\_lingkaran, jejari, dan nama termasuk besaran leksikal identifier variabel.

#### c. Operator

Operator aritmatika ( +, -, \*, / )

Operator logika (<, =, >, <=, >=, !=, <>)

Delimiter berguna sebagai pemisah atau pembatas,

contoh:

Karakter sebagai berikut ( ) { } ; . , :  
Karakter *white space*, antara lain:  
karakter spasi → kode ASCII 32  
karakter enter (*carriage return*) → kode ASCII 13  
karakter ganti baris (*line feed*)  
karakter penanda akhir file (*end of file*)

## **2. Definisi Syntax**

Menurut Firrar Utdirartatmo(2005:51) Syntax suatu bahasa pemrograman adalah satu himpunan peraturan yang menjelaskan bagaimana simbol-simbol bahasa dapat dirangkai bersama untuk membentuk pernyataan (statement) yang berarti.

Aturan formal yang mengatur bagaimana seseorang menulis instruksi yang valid dalam pada suatu bahasa.

## **3. Kesalahan Program**

Menurut Firrar Utdirartatmo (2005:77) menyatakan bahwa kompilator akan sering menemui program yang mengandung kesalahan, maka kompilator harus memiliki strategi apa yang harus dilakukan untuk menangani kesalahan-kesalahan tersebut. Kesalahan program bias merupakan:

### **a. Kesalahan Leksikal**

Misalkan kesalahan mengeja keyword, contoh:

*then* ditulis *ten*

### **b. Kesalahan sintaks**

Misalkan operasi aritmatika dengan jumlah kurung yang tidak pas,

contoh:  $A:=X+(B*(C+D)$