

BUKU RANCANGAN PENGAJARAN

Teknik Kompilator

dengan dukungan *e-Learning*

Disusun oleh:

Heru Suhartanto
heru@cs.ui.ac.id

Fakultas Ilmu Komputer
Universitas Indonesia
Desember 2007

Bab 1

INFORMASI UMUM

Nama mata ajar	:	Teknik Kompilator
Kode mata ajar	:	IKI 40800
Diberikan pada	:	Semester 7
Jumlah sks	:	4
Jenis sks	:	3 x 50 menit pembelajaran dikelas 1 x 50 menit pembelajaran mandiri 2 x 50 menit tugas kelompok
Prasyarat	:	Konsep Pemrograman, Struktur Data dan Algoritma, Toeri Automata dan Bahasa
Kaitan dengan mata ajar lain	:	
Dosen	:	Heru Suhartanto heru@cs.ui.ac.id
Tutor	:	

Petunjuk Perkuliahan

Pemelajaran Teknik Kompilator ini diselenggarakan dengan *blended method* antara interaktif tatap muka dan *e-Learning* dengan pendekatan *student-centered learning*. Oleh karena itu, istilah yang dipergunakan adalah pemelajaran (*learning*) bukan pembelajaran (*instruction*). Pada sesi *e-Learning*, mahasiswa mempelajari modul yang disediakan dan mengerjakan lembar kerja yang sesuai. Mahasiswa secara mandiri mempelajari materi dan berkolaborasi serta berinteraksi dengan sesama mahasiswa dan fasilitator melalui forum diskusi.

Sesi tatap muka dapat dibedakan dalam macam kegiatan, yang pertama pemelajaran materi baru dalam bentuk kuliah interaktif, maupun diskusi kelompok, dan yang kedua adalah tutorial, yang membahas kesulitan-kesulitan belajar, serta memberikan pengarahan dan sharing hal-hal khusus yang tidak mungkin disajikan secara elektronik, misalnya *sharing and pulling* untuk mendukung pemelajaran dengan *e-Learning*. Perlu difahami bahwa sesi ini tidak dapat disubstitusi dengan aktifitas lain,

dan bukan pengulangan pemberian materi pada sesi *e-Learning*. Pada kegiatan perkuliahanpun dilakukan secara interaktif dua arah, dan mahasiswa dituntut untuk secara aktif terlibat dalam aktivitas ini. Oleh karenanya, mahasiswa diharapkan untuk datang pada sesi ini.

Pada kedua mode pada *blended method* ini, mahasiswa diharapkan mempersiapkan diri terlebih dahulu dengan membaca sumber bacaan dan mempersiapkan *worksheet* (lembar kerja) yang sesuai. Suksesnya proses pembelajaran amat tergantung pada keaktifan mahasiswa.

Selama masa pembelajaran berlangsung, mahasiswa diharapkan untuk aktif mengerjakan lembar kerja yang diberikan. Pengerjaan lembar kerja selama proses pembelajaran bukan dimaksudkan hanya untuk melakukan latihan soal, namun lebih penting lagi, sebagai bagian proses membentuk pengetahuan (*construction of knowledge*) dan pendalaman (internalisasi).

Pertanyaan-pertanyaan pada lembar kerja sudah dirancang untuk menunjang proses pembelajaran. Mahasiswa yang sudah memahami tanpa perlu mengerjakan lembar kerja lebih lanjut dapat meneruskan proses pembelajaran tanpa harus mengerjakan keseluruhan pertanyaan satu demi satu. Secara singkat, selama pembelajaran mahasiswa diharapkan *ready to think*, dan *ready to work*, tidak sekedar menjadi pembaca atau pendengar untuk menjamin terjadinya proses pembelajaran yang efektif.

Pemelajar harus senantiasa memantau kemajuan belajarnya sendiri. Pengerjaan lembar kerja dapat dipergunakan untuk itu. Jika mengalami kesulitan dalam mengerjakan lembar kerja, harap segera menyampaikan di forum. Sebelum mengikuti pembelajaran, mahasiswa diharapkan untuk mempersiapkan diri dengan membaca materi dari buku acuan.

Keaktifan mahasiswa dalam berkontribusi dalam forum menjadi salah satu kriteria penilaian. Tidak hanya frekuensi terlibat di forum saja yang diperhatikan, melainkan mutu dari kontribusinya juga.

Bab 2

SASARAN PEMELAJARAN

Tujuan umum

Mata ajar ini mempersiapkan mahasiswa untuk dapat menyelesaikan masalah yang terkait dengan perancangan bahasa pemrograman dan teknik teknik pembentuk kompilator – program pemroses bahasa pemrograman.

Sasaran pemelajaran

Secara umum mahasiswa akan memahami komponen suatu kompilator, bagaimana interaksi antara komponen dan bagaimana pembuatan dan pengembangan komponen tersebut. Secara rinci, tujuan tersebut dijabarkan dalam sasaran pemelajaran terminal dan penunjang berikut ini.

Sasaran pembelajaran terminal

1. Apabila diberi suatu pola token, mahasiswa mampu membuat ekspresi regular dan membuat programnya secara efisien untuk mengenali suatu token.
2. Mahasiswa memahami tatabahasa (grammar) mampu membuat grammar dan beberapa tabel parsingnya serta mampu merancang suatu parser (pengurai) yang tidak ambigu dan membuat programnya secara efisien.
3. Mahasiswa mampu membuat komponen analisis semantik dan menggabungkannya dengan Parser menjadi suatu kesatuan bagian suatu kompilator.
4. Mahasiswa mampu memahami dan membuat Simbol Tabel untuk kebutuhan suatu kompilator.
5. Mahasiswa memahami alokasi storage dalam suatu kompilator dan mengimplementasikannya.
6. Mahasiswa mampu memahami bahasa mesin yang akan dipakai sebagai bahasa sasaran suatu kompilator.
7. Mahasiswa mampu menambah fitur fitur suatu kompilator dengan melengkapi komponen kompilator dengan hal hal yang diperlukan.
8. Mahasiswa mampu membuat kompilator simulasi yang dapat memroses bahasa sumber menjadi bahasa sasaran yang dapat dijalankan pada suatu computer.

Diagram alur pokok bahasan

Alur pokok pembahasan secara umum bersifat linier, yakni dari topik awal pengenalan kompilator sampai optimisasi suatu kompilator. Namun khusus beberapa topik seperti simbol tabel, pemeriksaan type, alokasi storage secara umum, perangkat pembantu pembentuk penganalisis leksikal dan parser dapat dilakukan secara parallel pada setengah semester pertama.

Bab 3

SUBPOKOK BAHASAN DAN RUJUKAN

No	Pokok Bahasan	Sub pokok Bahasan	Rujukan	Modul
	Pengenalan	Fungsi dan kegunaan teknik kompilator		
	Analisa Leksikal	Analisa Leksika	[1] 3.1-3.4; [2]	
	Analisa Sintak	Pengenalan Grammar	[1] 4.1-4.7	
		Grammar Model	[2]	
		Parsing Tree dan Derivasi		
		Parsing Prediktive		
		Parsing Shift dan Reduksi		
		Parsing SLR		
		Parsing LALR		
		UTS		
	Tabel Simbol	Tabel Linier dan Hash	[1] 7.6; [2]	
		Operasi pada Tabel Simbol		
	Analisa Semantik	Deklarasi Nama / Variabel	[1] 6.1-6.2; [2]	
		Operasi atribut – type checking		
		Contoh Analisis Semantik – Context Checker		
	Run-Time Environment		[1] 7.1-7.5; [2]	
	Teknik Alokasi Storage			
	Kode Sasaran	Kode Sasaran umum		
		Kode Sasaran Model	[2]	
	Kode Intermediate		[1] 8.1 – 8.3	
	Pembentukan Kode	Pembentukan Kode Umum	[1] 9.1-9.4	
		Prosedur dan Fungsi Tanpa Parameter		
		Prosedur dan Fungsi dengan Parameter		
	Optimisasi		[1]10.1-10.2	

Rujukan

Utama

1. Compilers: Principles, Techniques, and Tools
Aho, A. V., Sethi, R., Ullman, J. D. Addison Wesley 1986
2. Heru Suhartanto, Model Pemroses Bahasa Pemrograman Dengan tools berbasis Java, 2006. ,

Penunjang

3. Modern Compiler Implementation in Java
<http://www.cs.princeton.edu/~appel/modern/java/>
4. Sumantri Slamet dan Heru Suhartanto, Teknik Kompilasi, Elex Media Komputindo, Jakarta, 1993. – akan direvisi menjadi Teknik Kompilasi : Teori dan Simulasi, target draft Des 06.
5. Dick Grune et. Al, Modern Compiler Design, John Wiley & Sons, 2000
6. JLex
7. JavaCup

Bab 4

MATRIKS KEGIATAN

Metode pembelajaran:

1. Diskusi Interaktif (DI)
2. Belajar Mandiri (BM)
3. Kuliah Interaktif/ tatap muka (KI)

Sumber Pemelajaran

1. Buku Teks
2. *Handout*
3. Internet
4. Manual Java

Media Instruksional

1. Internet (I)
2. White board, infocus (Wbi)

Matriks Kegiatan Perkuliahan

Minggu	Sasaran Pemelajaran	Metoda Pemelajaran	Pokok Bahasan	Sub pokok Bahasan	Media/modul
1		BM	Pengenalan	Fungsi dan kegunaan teknik kompilator	I/1
1	1	KI	Analisa Leksikal	Konsep Analisis Leksikal, Model Analisis Leksikal	Wbi/1
2	2	BM	Analisa Sintak	Pengenalan Grammar	I/2
		Tugas mandiri	Tugas Kelompok, eksperimen model analisis leksikal dan sintak, due 1 week		
2	2	KI		Grammar Model	Wbi
3	2	BM		Parsing Tree dan Derivasi	
3	2	KI		Parsing Prediktive	Wbi
4	2	KI		Parsing Shift dan Reduksi	Wbi
4	2	KI		Parsing SLR	Wbi

4	2	KI		Parsing LALR	Wbi
5	2	BM		Parsing Operator Presedensi	
5	4	KI	Tabel Simbol	Tabel Linier dan Hash	Wbi
6	4	BM		Operasi pada Tabel Simbol	I
7		UTS			
7	3	KI	Analisa Semantik	Deklarasi Nama / Variabel	Wbi
7	3			Operasi atribut – type checking	
8	3	KI		Contoh Analisis Semantik – Context Checker	Wbi
9	5	BM	Run-Time Environment		I
10	5	KI	Teknik Alokasi Storage		Wbi
10			Tugas kelompok 2, implementasi prosedur / function tanpa parameter, due 2 minggu		
11	6	BM	Kode Sasaran	Kode Sasaran umum	I
11	6	KI		Kode Sasaran Model	Wbi
12	6	BM	Kode Intermediate		I
13	6	BM	Pembentukan Kode	Pembentukan Kode Umum	I
			Tugas kelompok 3, implementasi prosedur / fungsi dengan parameter, due 3 minggu		

14	7,8	KI		Prosedur dan Fungsi Tanpa Parameter	Wbi
14	7,8	KI		Prosedur dan Fungsi dengan Parameter	Wbi
14	7,8	BM	Optimisasi		I
15		BM		Review	
UAS					

Bab 5

CONTOH-CONTOH PERTANYAAN PENGARAH

Minggu	Soal	Ket
1, 2	1. Tentukan ekspresi reguler untuk suatu identifier 2.	
3	1. Apa dampak dari grammar yang ambigu 2.	

Bab 6

EVALUASI HASIL PEMELAJARAN

Bentuk/jenis instrumen

1. Tugas Individu/Kelompok (laporan,program, testing)
2. Kuis (isian singkat)
3. Ujian Tertulis (*essay*, jawaban singkat)

Skema Penilaian Akhir

No	Komponen	Bobot
1.	Tugas Individu/Kelompok (3-5)	50%
2.	Bonus (Kontribusi/ keaktifan)	10%
3.	Ujian Tengah Semester	25%
4.	Ujian Akhir Semester (komprehensif)	25%
	Total	110%

Kisi – Kisi Tugas

Dalam pengerjaan tugas, mahasiswa wajib mengumpulkan total minimum 40% dari pekerjaan berikut dimana tugas butir 2, 4 dan 5 wajib dilakukan:

1. [5%] Mandiri, Tugas mentest model kompilator dengan variasi program input standar yang harus mampu dilakukan oleh model kompilator. Mentest kemampuan model mendeteksi variasi kesalahan program input.
2. [10%] Kelompok, Tugas membuat bahan e-learning with flash and voice untuk materi pembelajaran Belajar Mandiri – lihat Bab 4 Matriks kegiatan.
3. [5%] Mandiri, Tugas membuat terjemahan suatu program ke dalam bahasa mesin simulasi kuliah.
4. [10%] Kelompok, Tugas membuat laporan riset berbasis survei tentang status terkini teknologi Teknik Kompilator. Status terkini dapat dikaitkan dengan perkembangan ICT saat ini seperti munculnya General Purpose Programming on GPU, Distributed/Cluster Computing/Clouds Computing.
5. [10%] Kelompok, Tugas pelengkapan model dengan prosedur dan fungsi tanpa parameter. Modul yang perlu direvisi adalah modul tabel simbol guna melengkapi atribut program counter yang harus ada pada prosedur dan fungsi, modul analisis semantik atau context checker yang harus dilengkapi dengan implementasi aturan yang terkait, dan modul code generator yang harus dilengkapi dengan aturan code generator yang terkait.

6. [10%] Kelompok, Tugas pelengkapan model dengan prosedur dan fungsi dengan parameter. Modul yang perlu direvisi adalah modul tabel simbol guna melengkapi attribute untuk jumlah parameter, dan pointers ke parameter parameter prosedur dan fungsi. Juga modul analisis semantik atau context checker yang harus dilengkapi dengan implementasi aturan yang terkait, dan modul code generator yang harus dilengkapi dengan aturan code generator yang terkait.

Skema penilaian Tugas

1. Fungsi dan Prosedur tanpa parameter
 - a. 10 %, dokumentasi
 - b. 10 %, kompilasi
 - c. 15 %, variasi testing yang masing-masing terdiri dari 2 point untuk prosedur sederhana, fungsi sederhana, fungsi non sederhana, fungsi recursive, prosedur recursive, prosedur dalam fungsi atau fungsi dalam prosedur, dan 3 point untuk prosedur atau fungsi nested
 - d. 55% , eksekusi variasi testing 7 point utk prosedur sederhana, 7 point untuk fungsi sederhana, 10 point untuk fungsi non sederhana, 10 point untuk fungsi recursive, 10 point untuk prosedur recursive, 10 point untuk prosedur dalam fungsi atau fungsi dalam prosedur, dan 11 point untuk prosedur atau fungsi nested
2. Fungsi dan Prosedur dengan parameter
 - a. 12 % , dokumentasi
 - b. 88 %, eksekusi variasi testing, yang terdiri dari 12 test input benar, 6 test input salah, dan 4 test spesial.

Kisi-kisi naskah Ujian

1. Definisi dan contoh pola dalam analisis leksikal
2. membuat himpunan first dan follow berdasarkan grammar yang diberikan
3. memeriksa apakah suatu input program dapat diproses dengan berbagai jenis parser
4. membuat bagian bagian (himpunan state, diagram state, tabel parser) berbagai jenis parser.
5. membuat diagram tabel simbol yang dapat terbentuk berdasarkan input program
6. menjelaskan skema alokasi storage
7. menjelaskan run time environment
8. menuliskan kode target berdasarkan program input
9. menjelaskan aksi kode generator yang terkait dengan proses pemakaian prosedur dan fungsi tanpa dan dengan parameter
10. mengidentifikasi blok dasar dalam suatu kode intermediate dan melakukan optimisasi yang diperlukan.

Bab 7

CONTOH-CONTOH SOAL

Contoh Soal Ujian Akhir

Pertanyaan singkat

- tuliskan bahasa mesin untuk ekspresi $a := b + c$, anda dapat memberikan asumsi tentang lexical level dan order number variable yang terlibat.
- Apa saja yang harus dilakukan oleh kode generator pada saat proses mendekati deklarasi suatu prosedur atau function?
- Apa saja yang harus dilakukan oleh kode generator pada saat proses akan mengeksekusi pemanggilan suatu prosedur atau function.
- Apa saja yang harus dilakukan oleh kode generator pada saat proses meninggalkan pemanggilan suatu prosedur atau fungsi

Pertanyaan Essays

- Diberikan program sumber seperti berikut

1	{ var a: int
2	var b: int
3	--
4	---
5	proc c(a :int, b:int)
6	{
7	--
8	--
9	}
10	Var e : int
11	--
12	proc d(a:int, b:int)
13	{
14	--
15	--
16	}
17	Var f: int
18	--
19	}

Gambarkan keadaan tabel symbol pada baris 3, pada akhir baris 5, pada baris akhir 9, pada akhir baris 12 dan pada akhir baris 17

- Berikut adalah Kode intermediate suatu source code, ada berapa blok dasar? gambarkan kaitan blok dasar, lakukan optimisasi yang diperlukan.

(1) $i := m - 1$	(16) $t_7 := 4 * i$
(2) $j := n$	(17) $t_8 := 4 * j$
(3) $t_1 := 4 * n$	(18) $t_9 := a[t_8]$
(4) $v := a[t_1]$	(19) $a[t_7] := t_9$
(5) $i = i + 1$	(20) $t_{10} := 4 * j$
(6) $t_2 := 4 * i$	(21) $a[t_{10}] := x$
(7) $t_3 := a[t_2]$	(22) goto (5)
(8) if $t_3 < v$ goto (5)	(23) $t_{11} := 4 * i$
(9) $j := j - 1$	(24) $x := a[t_{11}]$
(10) $t_4 := 4 * j$	(25) $t_{12} := 4 * i$
(11) $t_5 := a[t_4]$	(26) $t_{13} := 4 * n$
(12) if $t_5 > v$ goto (9)	(27) $t_{14} := a[t_{13}]$
(13) if $i >= j$ goto (23)	(28) $a[t_{12}] := t_{14}$
(14) $t_6 := 4 * i$	(29) $t_{15} := 4 * n$
(15) $x := a[t_6]$	(30) $a[t_{15}] := x$