LAPORAN TUGAS AKHIR

Implementasi Algoritma *Principal Type* dan Algoritma *Type Inhabitant* dari *Type Theory* TA-Lambda Menggunakan *Definite Clause Grammar* dalam PROLOG

Oleh:

Ario Santoso

1204007038

FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA

DEPOK

2008

Halaman Persetujuan

Judul Tugas Akhir :

Implementasi Algoritma *Principal Type* dan
Algoritma *Type Inhabitant* dari *Type Theory* TA-Lambda
Menggunakan *Definite Clause Grammar* dalam PROLOG

Nama : Ario Santoso

NPM : 1204007038

Laporan Tugas Akhir ini telah diperiksa dan disetujui.

Depok, Juli 2008

L.Y. Stefanus

Pembimbing Tugas Akhir

Abstrak

Studi tentang *type theory* telah memberikan kontribusi penting dalam dunia ilmu komputer, terutama dalam rekayasa perangkat lunak, basis data, *computational linguistics*, desain bahasa pemrograman, *automated theorem proving*, *high performance compiler* dan keamanan jaringan komputer. Tugas akhir ini berfokus pada varian *type theory* yang disebut *Type Assignment* (TA). Kontribusi dari tugas akhir ini terdiri dari tiga hal pokok. Pertama, algoritma *Principal Type* (PT) dan pencarian *type inhabitant* diimplementasikan dalam PROLOG dengan menggunakan Definite Clause Grammar (DCG). Hasil implementasi ini dapat dipakai untuk mencari tipe dari sebuah λ-*term* dan juga sebaliknya, mencari *inhabitant* (berupa λ-*term*) dari sebuah tipe. Kedua, seluk-beluk TA, terutama algoritma PT dan pencarian *type inhabitant*, dipaparkan dengan bahasa yang lebih mudah dimengerti dibandingkan literatur yang sudah ada. Ketiga, sebuah antarmuka grafis dibangun untuk memudahkan *user* dalam menggunakan (mencoba) kedua algoritma tersebut. Dengan demikian, *software* ini bisa digunakan sebagai *test bed* untuk mempelajari TA, maupun untuk bereksperimen dalam *type theory*.

xv + 184 hal.; 31 gambar.; 1 tabel.; Lampiran A-P; Bibliografi: 13 (1927-2007)

Kata Pengantar

Puji syukur penulis panjatkan ke hadirat Allah SWT karena atas rahmat dan pertolongan-Nya-lah penulis dapat menyelesaikan tugas akhir ini. Laporan ini berisi penjabaran tentang tugas akhir yang dilakukan oleh penulis.

Dalam kesempatan ini penulis juga hendak menyampaikan rasa terima kasih yang sebesar - besarnya kepada semua pihak yang telah banyak membantu penulis selama pelaksanaan tugas akhir ini, dan juga selama penulis menempuh pendidikannya di Fakultas Ilmu Komputer Universitas Indonesia, terutama kepada:

1. Keluarga penulis yang telah memberikan dukungan serta doa.
2. Bapak L.Y. Stefanus, selaku pembimbing tugas akhir penulis, yang senantiasa membimbing penulis dalam mengerjakan tugas akhir ini.
3. Bapak T.Basaruddin, selaku pembimbing akademik penulis, yang senantiasa membimbing penulis selama mengenyam pendidikan di Fakultas Ilmu Komputer Universitas Indonesia.
4. Semua dosen Fakultas Ilmu Komputer Universitas Indonesia, yang telah membimbing penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Indonesia.
5. Pembuat LaTeX, yang memungkinkan saya dapat menulis dokumen laporan tugas akhir ini dengan nyaman.
6. Teman-teman lab 1237, Alidz, Arya, Pandu, dan Aji. Terima kasih atas segala dukungannya.
7. Aniz, Rama, Akin, Imam, Satrio, selaku teman seperjuangan penulis dalam mengerjakan skripsi. Walaupun kita semua berbeda bidang, tapi kita tetap punya

satu tujuan. Special thanks untuk Dewi K.S.

8. Rahmad Mahendra, selaku teman penulis yang senantiasa mengajarkan teknik-teknik penulisan kepada penulis, dan juga teman berdiskusi penulis.

9. Arfan, Desmond, Mimi, Eliza dan Franky, selaku rekan penulis yang menjadi teman berdiskusi dan memberikan masukan kepada penulis.

10. Andreas dan Rozie 2003, selaku teman penulis yang mengajarkan menggunakan LaTeX dan membantu mencari solusi ketika terjadi permasalahan berkaitan dengan LaTeX.

11. Richard, Wahyu Mirza, Daniel Cahyadi, Rado, Angky, dan Albert, selaku teman-teman penulis yang selalu memberikan dukungannya.

12. Rekan - rekan SUN Microsystem, komunitas Java User Group Indonesia (JUGI) dan komunitas Open Solaris User Group Indonesia (OSUGI), khususnya pada Pak Harry, Pak Aan, Pak Adhari, Pak Junus, Thomas W., Joshua, dan Mbak Efni.

13. Semua teman-teman fasilkom UI khususnya angkatan 2004 untuk dukungannya dalam pengerjaan tugas akhir ini.

14. Semua pihak yang telah membantu dan mendukung penulis dalam penyelesaian Tugas Akhir ini.

Penulis sadar bahwa Laporan ini masih memiliki berbagai kekurangan dan keterbatasan. Oleh karena itu penulis menerima dengan tangan terbuka semua kritik dan saran terhadap Laporan ini. Namun demikian, penulis berharap semoga Laporan ini bisa bermanfaat bagi pembaca dalam memberikan wawasan baru.

Depok, Juli 2008

Ario Santoso

Daftar Isi

Daftar Gambar

Daftar Tabel

Bab 1

Pendahuluan

Bab ini memaparkan latar belakang, tujuan, ruang lingkup, metodologi penelitian, dan sistematika penulisan laporan tugas akhir ini.

1.1 Latar Belakang

Misalkan A adalah sebuah himpunan yang mengandung semua himpunan yang bukan merupakan elemen dari dirinya sendiri. Apakah A merupakan elemen dari dirinya sendiri (himpunan A tersebut)? Atau mungkin untuk lebih mudah memahaminya, kita tinjau pernyataan berikut: "Seorang tukang cukur mencukur semua orang yang tidak mencukur dirinya sendiri, apakah tukang cukur tersebut mencukur dirinya sendiri?". Hal tersebut merupakan paradoks Russel yang paling terkenal, yang muncul sekitar tahun 1900an [Mun07] [BR27]. Baik jawaban positif atau negatif pada pertanyaan tersebut akan menghasilkan kontradiksi.

Dengan tujuan untuk mengatasi paradoks Russel yang telah mengacaukan fondasi matematika pada masa tersebut, diajukanlah *type theory*. Kemudian seiring berkembangnya waktu, penggunaan *type theory* diperluas hingga menjadi alat standar yang digunakan para ahli logika (*logician*), terutama dalam bidang *proof-theory*. Pada tahun 1970an, kebutuhan akan bahasa pemrograman yang lebih tangguh membuat *type theory* menjadi perhatian orang-orang dari dunia ilmu komputer (*computer scientist*). Kemudian beberapa bahasa pemrograman baru dikembangkan pada tahun 1970an dan 80an dengan didasarkan pada *type theory*. Contoh utamanya adalah ML, dikembangkan oleh universitas Edinburgh, melalui grup yang dipimpin oleh Robin Milner. Contoh

yang lainnya adalah HOL (Universitas Cambridge), Miranda (*Research Software Ltd.*) dan Nuprl (Universitas Cornell)[Hin97].

Studi tentang *type system* dan aplikasinya telah memberikan kontribusi yang penting dalam dunia ilmu komputer, seperti dalam rekayasa perangkat lunak, basis data, *computational linguistics*, desain bahasa pemrograman, *automated theorem proving*, *high performance compiler* dan keamanan jaringan komputer. Dalam ilmu komputer terdapat dua cabang utama dalam mempelajari *type system*, yaitu:

1. Aliran yang lebih memfokuskan pada sisi praktis dari *type system* yang menyangkut aplikasinya pada bahasa pemrograman.

2. Aliran yang memfokuskan pada hal yang lebih abstrak, yaitu mempelajari hubungan antar berbagai *type system* dan beraneka ragam logika melalui *Curry Howard correspondence*.

Type system pertama kali digunakan dalam dunia ilmu komputer sekitar awal tahun 1950 pada bahasa pemrograman seperti FORTRAN. Penggunaan konsep ini berperan untuk meningkatkan efisiensi proses perhitungan, dengan cara membedakan antara penghitungan bilangan bulat (integer) dan bilangan real. Pembedaan ini memungkinkan *compiler* untuk menggunakan representasi yang berbeda dan membuat instruksi mesin yang sesuai untuk operasi primitif. Selain itu, *type system* juga telah berperan banyak dalam desain bahasa pemrograman. Hal yang paling dapat terlihat jelas adalah memungkinkannya deteksi dini terhadap beberapa *programming error* [Pie02].

Type theory memiliki banyak *varian*, salah satunya adalah *Type Assignment* (TA). TA merupakan salah satu *type theory* yang paling sederhana. Dalam sistem TA, sebuah tipe hanya tersusun dari *type-variable* dan tanda panah. *Term* dalam sistem TA ini mengikuti λ-*calculus* yang *term*-nya hanya tersusun dari λ-*abstraction*, *application* dan *term-variable*. Kemampuan TA dalam mengekspresikan sesuatu (*expressive power*) mendekati sistem yang disebut *simple type theory*, suatu sistem yang dikembangkan oleh Alonzo Church [Hin97].

Pemikiran dan algoritma di dalam TA tidaklah *trivial*, dan banyak teknik kompleks yang digunakan untuk menganalisa struktur dalam *type theory* yang lebih kompleks hadir di TA dalam bentuk yang mudah untuk dipahami. Hal ini menjadikan TA sebagai suatu sarana pembelajaran yang baik untuk mempelajari teknik dalam *type*

theory secara keseluruhan. Dengan mempelajari teknik dasar dari *type theory* melalui TA, kita juga akan mendapatkan fondasi yang baik untuk mempelajari *type system* yang lain [Hin97].

Melihat pentingnya TA dan juga kaitannya dengan *type theory*, dalam tugas akhir ini penulis mempelajari TA dengan berfokus pada dua algoritma dalam TA, yaitu algoritma *Principal Type* dan algoritma pencarian *type inhabitant*. Algoritma *Principal Type* merupakan algoritma untuk memberikan tipe pada sebuah λ-*term*. Algoritma ini menerima *input* sebuah λ-*term*. Jika λ-*term* yang diberikan dapat memiliki tipe (*typable*), maka algoritma ini akan mengeluarkan tipe paling umum dari λ-*term* tersebut. Namun jika λ-*term* yang diberikan tidak dapat memiliki tipe (*untypable*), maka algoritma ini akan mengeluarkan pernyataan yang benar bahwa λ-*term* tersebut tidak dapat memiliki tipe. Kemudian algoritma pencarian *type inhabitant* merupakan algoritma untuk mencari *inhabitant* dari sebuah tipe. Algoritma ini merupakan bagian inti dari algoritma penghitungan *type inhabitant* Ben-Yelles, yang berfungsi untuk menentukan apakah suatu tipe τ memiliki sejumlah tak hingga atau berhingga *inhabitant*, atau mungkin tidak memiliki *inhabitant* sama sekali.

Dalam tugas akhir ini kedua algoritma tersebut diimplementasikan menggunakan PROLOG, dan hasilnya dijelaskan pada laporan ini. Dalam mengimplementasikan kedua algoritma tersebut, penulis juga mengembangkan *parser* untuk λ-*term* dan tipe menggunakan *Definite Clause Grammar* (DCG) dalam PROLOG.

1.2 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Mempelajari algoritma *Principal Type* (PT *algorithm*) dan juga algoritma pencarian *type-inhabitant* pada sistem *Type-Assignment* untuk λ-*calculus* (sistem TA_λ), serta memberikan pemaparan yang diharapkan dapat lebih mudah dimengerti dibandingkan dengan literatur yang sudah ada.

2. Membuat *parser* λ-*term* dengan menggunakan DCG dalam PROLOG.

3. Mengimplementasikan algoritma *Principal Type*.

4. Membuat *parser* untuk tipe dengan menggunakan DCG dalam PROLOG.

5. Mengimplementasikan algoritma pencarian *type-inhabitant*.

6. Membuat antarmuka yang memadai untuk memudahkan *user* dalam menggunakan (mencoba) kedua algoritma tersebut.

1.3 Ruang Lingkup

Cakupan tugas akhir ini meliputi pembelajaran algoritma *principal type* dan juga algoritma pencarian *type inhabitant* pada sistem TA_{λ}. Dalam tugas akhir ini, dua algoritma tersebut diimplementasikan. Selain itu, tugas akhir ini juga meliputi pembuatan *parser* untuk λ-*term* dan tipe, dengan menggunakan *Definite Clause Grammar* (DCG) dalam PROLOG, serta pembuatan antarmuka yang memadai untuk memudahkan *user* dalam menggunakan (mencoba) kedua algoritma tersebut.

1.4 Metodologi Penelitian

Penelitian yang dilakukan terdiri dari beberapa tahapan sebagai berikut:

1. Studi literatur. Pembelajaran literatur melalui buku dan artikel yang berhubungan dengan subyek yang sedang dibahas.

2. Implementasi. Implementasi algoritma *Principal Type* dan algoritma pencarian *type inhabitant* yang dilakukan dengan PROLOG. Implementasi *parser* untuk λ-*term* dan tipe dari λ-*term* dilakukan dengan DCG dalam PROLOG, dan pembuatan antarmuka menggunakan Java.

3. Uji coba hasil implementasi. Melakukan uji coba penerapan algoritma *Principal Type* pada sejumlah λ-*term* dan algoritma pencarian *type inhabitant* pada sejumlah tipe.

4. Analisa terhadap semua hal yang telah dilakukan dalam proses penelitian. Analisa ini menghasilkan sejumlah kesimpulan dan saran.

1.5 Sistematika Penulisan Laporan

Berikut adalah sistematika penyajian laporan ini:

Bab 1 Pendahuluan. Bab ini memaparkan latar belakang, tujuan, ruang lingkup, metodologi penelitian, dan sistematika penulisan laporan tugas akhir.

Bab 2 *Definite Clause Grammar* (DCG). Bab ini menjelaskan secara sekilas tentang *Definite Clause Grammar* (DCG) dalam PROLOG. Dalam bab ini juga disertakan beberapa contoh penggunaan DCG.

Bab 3 *Type-Free* λ*-calculus*. Bab ini menjelaskan pengertian dan beberapa konsep terkait *type-free* λ*-calculus*.

Bab 4 *Type-Assignment* pada λ*-term* (TA_λ). Bab ini memberikan pengenalan kepada sistem TA_λ.

Bab 5 *Principal-Type* (PT) *Algoritm*: Algoritma Pemberian Tipe pada λ*-term*. Bab ini menjelaskan algoritma *principal type*. Dalam bab ini juga dijelaskan beberapa konsep pendukung yang dibutuhkan untuk memahami algoritma tersebut.

Bab 6 Implementasi Algoritma *Principal Type*. Bab ini menjelaskan implementasi algoritma *principal type*, yang juga disertai implementasi algoritma unifikasi Robinson. Selain itu bab ini juga menjelaskan implementasi *parser* untuk λ*-term* yang diimplementasikan dengan DCG dalam PROLOG.

Bab 7 *Typed Term*. Bab ini menjelaskan *typed term*. Alih-alih menjelaskan *typed term* sebagai pendekatan TA_λ yang dilakukan Church secara keseluruhan, bab ini hanya menjelaskan *typed term* sebagai notasi alternatif dalam deduksi TA_λ.

Bab 8 Algoritma Pencarian *Type Inhabitant*. Bab ini memaparkan algoritma pencarian *type inhabitant*.

Bab 9 Implementasi Algoritma Pencarian *Type Inhabitant*. Bab ini menjelaskan implementasi algoritma pencarian *type inhabitant*. Selain itu bab ini juga menjelaskan implementasi *parser* untuk tipe yang diimplementasikan dengan DCG dalam PROLOG.

Bab 10 Kesimpulan dan Saran.