



PEMELIHARAAN AGREEMENT DENGAN  
MENGUNAKAN PERUBAHAN SKEMA DAN  
ONTOLOGI PADA LINGKUNGAN PEER-TO-PEER (P2P)

DISERTASI

Untuk Memenuhi Salah Satu Syarat Meraih Gelar Doktor Teknologi Informasi  
di bawah Pimpinan Rektor Universitas Gunadarma  
Professor Doktor E.S. Margianti, SE, MM

Dipertahankan dalam Sidang Terbuka Senat Universitas Gunadarma  
Pada Hari Selasa Tanggal 30 Oktober 2007 Jam 09.00 - 12.00 WIB

LINTANG YUNIAR BANOWOSARI  
99203202

PROGRAM DOKTOR TEKNOLOGI INFORMASI  
UNIVERSITAS GUNADARMA  
2007

LOGO GUNADARMA ONLY  
GOLD COLOR  
EMBOSS PRINTING

## ABSTRAK

Penelitian ini mengembangkan sebuah metode pemeliharaan *semantic agreement* berdasarkan *semantic distance* dengan menghitung perubahan skema lokal atau ontologi. Pendekatan ini sangat perlu dalam lingkungan yang dinamis dan otonom, dimana pendekatan yang ada berasumsi dalam pembuatan agreement / mapping dalam sebuah model statis. Disadari pada kondisi dinamis, monitoring secara manual adalah tidak memadai.

Kontribusi dari penelitian ini adalah suatu framework berbasis pendekatan pemeliharaan *semantic agreement* untuk lingkungan P2P. Framework ini didasarkan atas arsitektur P2P model hybrid dua level yang terdiri dari dua tipe peer: (1) super peer yang digunakan untuk me-register dan mengelola peer lainnya, dan (2) simple peer yang isinya diekspor dan dibagi dengan peer yang lain. Penelitian ini mengembangkan suatu model untuk pemeliharaan *semantic agreement* pada lingkungan P2P, sehingga pada pendekatan yang ada sebelumnya tidak terdapat mekanisme untuk mengetahui perubahan karena dalam kondisi statis akan menjadi kondisi dinamis. Dua inti utama adalah bagaimana menghitung perubahan local schema atau common ontologi dan hasil dari perhitungan digunakan untuk menentukan algoritma dalam memelihara agreement.

Sebuah uji coba dasar pada domain pencarian tenaga kerja telah dilakukan untuk memperlihatkan seberapa jauh capaian hasil pendekatan ini. Dari uji coba didapatkan hasil utama adalah (i) semakin banyak perubahan maka nilai *F - measure* semakin cenderung menurun, (ii) tidak terjadi perbedaan yang signifikan dalam *F - measure* untuk berbagai modifikasi (Add, Delete, Rename), dan (iii) pemilihan algoritma yang tepat akan membengaruhi *F - measure*.

Kata Kunci : *semantic agreement, ontologi, semantic web, P2P*

PEMELIHARAAN AGREEMENT DENGAN  
MENGUNAKAN PERUBAHAN SKEMA DAN  
ONTOLOGI PADA LINGKUNGAN PEER-TO-PEER

DESERTASI

LINTANG YUNIAR BANOWOSARI

Telah disetujui oleh :

**Profesor Dr. Suryo Guritno, M.Stat**

Promotor

**Dr.rer.nat. A. Benny Mutiara, S.Si, S.Kom, Dipl.Phys**

Ko-Promotor

**Dr. I Wayan Simri Wicaksana, S.Si.,M.Eng.**

Ko-Promotor

Jakarta, 2007

Judul desertasi	PEMELIHARAAN AGREEMENT DENGAN MENGUNAKAN PERUBAHAN SKEMA DAN ONTOLOGI PADA LINGKUNGAN PEER-TO-PEER
Nama mahasiswa	Lintang Yuniar Banowosari, S.Kom, M.Sc
Nomer pokok mahasiswa	99203202
Komite Pembimbing	
Promotor	Profesor Dr. Suryo Guritno, M.Stat
Ko-Promotor	Dr.rer.nat. A. Benny Mutiara, S.Si, S.Kom, Dipl.Phys
Komisi Pembimbing	Dr. I Wayan Simri Wicaksana, S.Si.,M.Eng.
Komite Penguji	
Ketua	Profesor Dr. Suryo Guritno, M.Stat
Anggota	Drs. Edi Winarko, M.Sc, PhD Prof. Dr. Michel Paindavoine Prof. Dr. E.S. Margianti, SE, MMSi Prof. Dr. Yuhara Sukra, M.Sc Prof. Suryadi Harmanto, S.Si, MMSi Prof. Dr Djati Kerami Dr. Prihandoko Dr.rer.nat. A. Benny Mutiara, S.Si, S.Kom, Dipl.Phys Dr. I Wayan Simri Wicaksana, S.Si.,M.Eng.

## RIWAYAT HIDUP

Nama	Lintang Yuniar Banowosari
Tempat dan Tanggal Lahir	Solo, 03-Juni-1968
S1	S.Kom.,Manajemen Informatika,STMIK Gunadarma, Jakarta,1992
S2	M.Sc.,Computer Science,Asian Institute of Technology, Bangkok,1994

## UCAPAN TERIMA KASIH

Alhamdulillah, Puji syukur penulis panjatkan kepada Allah SWT, karena dengan karunia-Nya lah penulis dapat menyelesaikan disertasi ini dengan lancar.

Dalam kesempatan ini, penulis mengucapkan terimakasih kepada Ibu Prof.Dr. E.S. Margianti., SE.,MM., Rektor Universitas Gunadarma, Bapak Dr. Djati Kerami, Ketua Program Strata 3 Teknologi Informasi Universitas Gunadarma, Bapak Dr. Prihandoko, sekretaris Program Strata 3 Teknologi Informasi Universitas Gunadarma.

Terimakasih kepada bapak Profesor Suryadi H.S, SSI, MMSI, selaku Promotor, yang telah mengarahkan penulis dan bapak Dr.rer.nat. A. Benny Mutiara, SSI, SKom, Dipl.Phys, selaku Ko-Promotor, yang telah membimbing dan memberikan masukan kepada penulis. Terimakasih juga kepada Prof.Dr. Kokou Yetongon dan Prof Dr. Nadine Cullot, Universite de Bourgogne Dijon, Perancis, yang memberikan wawasan mengenai ontologi dan semantic web.

Kesempatan ini pula penulis memberikan penghargaan yang sangat tinggi serta ucapan terimakasih yang tulus kepada bapak Dr. I Wayan Simri Wicaksana, S.Si, M.Eng, selaku pembimbing, yang telah membimbing, mengarahkan dan memberikan dukungan semangat serta kontribusi materi terbesar dalam penulisan disertasi ini, terimakasih Pak.

Kepada sekretariat program S3 Teknologi Informasi Universitas Gunadarma, kepada teman teman di Program S3 Teknologi Informasi Universitas Gunadarma terimakasih atas bantuan dan dukungannya. Serta terimakasih kepada semua pihak, yang tidak dapat penulis sebutkan satu persatu, yang dengan satu dan lain cara memberikan bantuan, dukungan dan doa kepada penulis.

Terimakasih untuk Bapak Suradjin dan Ibu Sri Wiyati, orangtua penulis yang selalu mendoakan dan mendukung penulis untuk selalu maju, dan terakhir, yang paling bersemangat mendukung penulis untuk menyelesaikan disertasi ini, suami tercinta, Teuku Yunufa, dan anak-anak terkasih, Teuku Aldefa dan Teuku Akelfa, terimakasih atas cinta, pengertian, dukungan dan doanya. Disertasi ini tidak akan mungkin ada tanpa dukungan dari kalian. I love you.

# Daftar Isi

Abstrak . . . . .	iii
Lembar Persetujuan . . . . .	iv
Riwayat Hidup . . . . .	vi
Ucapan Terimakasih . . . . .	vii
Daftar Isi . . . . .	x
Daftar Gambar . . . . .	xi
<b>I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	2
1.2 Perumusan Penelitian . . . . .	4
1.3 Gambaran Umum . . . . .	7
1.3.1 Arsitektur . . . . .	8
1.3.2 Mediasi Semantik . . . . .	9
1.3.3 Pemrosesan Kueri . . . . .	10
1.3.4 Kueri . . . . .	11
1.4 Kontribusi . . . . .	11
1.5 Organisasi Desertasi . . . . .	12
<b>II TINJAUAN PUSTAKA</b>	<b>13</b>
2.1 Semantic Agreement . . . . .	13
2.1.1 Pencocokan Skema dan Pemetaan Ontologi . . . . .	15
2.1.2 Manajemen Perubahan dalam Ontologi . . . . .	15
2.2 Peer-to-Peer . . . . .	22
2.2.1 Definisi dan Karakteristik P2P . . . . .	22
2.2.2 Peer-to-Peer Discovery . . . . .	25
2.2.2.1 Pencarian dalam P2P yang Tidak Terstruktur . . . . .	27
2.2.2.2 Pencarian dalam P2P yang Terstruktur . . . . .	28
2.2.2.3 Pencarian dalam Non-hierarchical DHT P2P . . . . .	29
2.2.2.4 Pencarian dalam Hierarchical DHT P2P . . . . .	29

2.2.2.5	Pencarian dalam Non-DHT P2P . . . . .	29
2.2.2.6	Pencarian dalam Loosely Structured P2P . . . . .	30
2.2.3	Peng-indeks-an dalam P2P . . . . .	30
<b>III PERTIMBANGAN PEMELIHARAAN AGREEMENT</b>		<b>33</b>
3.1	Pendahuluan . . . . .	33
3.2	Fitur Pendekatan Semantic Agreement . . . . .	38
3.2.1	Penemuan dan Pencocokan dalam Lingkungan P2P . . . . .	38
3.2.2	Model untuk Merepresentasikan Informasi Peer . . . . .	40
3.3	Isu Arsitektur Pendekatan Semantic Agreement . . . . .	41
3.3.1	Lingkungan Berbagi Informasi . . . . .	42
3.3.2	Pengukuran Similaritas . . . . .	44
3.3.2.1	Pencocokan Label . . . . .	45
3.3.2.2	Similaritas Menggunakan Struktur Konsep . . . . .	47
3.4	Semantic Agreement . . . . .	50
3.4.1	Unit Agreement . . . . .	51
3.4.2	Pengembangan dari Unit Agreement . . . . .	51
3.4.3	Pemrosesan Kueri Menggunakan Unit Agreement . . . . .	53
3.5	Peer Bergabung, Peer Meninggalkan dan Penemuan Peer . . . . .	55
3.5.1	Peer Bergabung dan Meninggalkan . . . . .	55
3.5.2	Penemuan Peer . . . . .	55
3.6	Isu Perubahan Versi Ontologi dan Skema Lokal . . . . .	56
3.6.1	Manajemen Perubahan Versi Ontologi . . . . .	57
3.7	Ilustrasi Contoh . . . . .	58
<b>IV Hasil Pengujian Pendekatan</b>		<b>72</b>
4.1	Pendahuluan . . . . .	72
4.2	Persiapan Pengujian Sederhana . . . . .	72
4.3	Pelaksanaan Pengujian Sederhana . . . . .	72
4.4	Hasil dan Diskusi . . . . .	74
<b>V Kesimpulan dan Rencana Ke Depan</b>		<b>78</b>
5.1	Umum . . . . .	78
5.2	Tujuan Penelitian . . . . .	78
5.3	Hasil . . . . .	79
5.4	Kontribusi . . . . .	79
5.5	Rencana Ke Depan . . . . .	79



# Daftar Gambar

1.1	Konten Peer (Bagian dari Ontologi dan Skema Ekspor)	6
1.2	Komponen dari <i>Peer Agreement Approach</i>	8
2.1	Klasifikasi Jaringan P2P)	25
2.2	Klasifikasi Pencarian di P2P)	27
2.3	Klasifikasi Indexing di Jaringan P2P)	31
3.1	Contoh P2P Marvick	36
3.2	Penemuan dan Pencocokan di Lingkungan P2P [Wicaksana, 2006])	39
3.3	Representasi Model <i>Agreement</i> )	43
3.4	Representasi <i>Unit Agreement</i> )	52
3.5	Langkah dari Pengembangan <i>Unit Agreement</i> )	53
3.6	Skema Ekspor dari Peer PP1 dan PP2 untuk contoh layanan Job Matching	59
3.7	Alur Pemeliharaan Agreement	61
3.8	Proses Pemilihan Kamus On-line)	67
3.9	Evaluasi Pencocokan Label Pada Bahasa Indonesia	68
3.10	Bagian dari Common Ontology - HR-XML	69
3.11	Kelas Ragu Ragu	70
3.12	UpperClass dari Kelas Ragu Ragu	71
4.1	Modifikasi <i>Add, Delete dan Rename</i> pada 10 LS	75
4.2	Skema Lokal untuk Masing-masing Skenario	76
4.3	Rangkuman Seluruh Skenario Modifikasi untuk 10 LS	76

# Bab I

## PENDAHULUAN

Sebagian besar organisasi dalam membuat perencanaan kebijaksanaan dan pengambilan keputusan berdasarkan informasi yang diambil atau dikumpulkan dari berbagai macam sumber. Sumber-sumber tersebut kadang kala merepresentasikan informasi pada model yang berbeda beda, termasuk tingkat yang paling tinggi sampai yang paling rendah dari organisasi tersebut. Setiap tingkatan mempunyai kebutuhan yang berbeda dalam jenis, model dan representasi untuk menggambarkan informasi yang dimiliki. Perbedaan ini merupakan isu menarik dalam pengaksesan informasi dari beragam sumber.

Interoperabilitas sistem informasi sudah sangat luas dipelajari dalam komunitas penelitian keterpaduan data tradisional (traditional data integration) dan sudah banyak isu yang berhubungan dengan resolusi konflik data sudah ditemukan, dengan menggunakan metodologi dan pendekatan yang berbeda. Pendekatan tersebut beragam, mulai dari database translation, database federation sampai dengan semantic data mediation berdasarkan wrapper dan arsitektur mediator. Telah ada beberapa usaha memperluas atau melanjutkan penelitian pendekatan interoperabilitas yang mengatur resolusi konflik dalam data dan aplikasi yang kompleks seperti sistem informasi pemerintahan elektronik (e-governement), menghasilkan beberapa definisi standar.

Beberapa studi menekankan perlunya untuk memecahkan konflik data pada tingkat yang lebih tinggi dari suatu abstraksi di mana menekankan arti yang berasosiasi dengan data. Sehingga resolusi itu dapat secara tepat direpresentasikan dan digunakan untuk membuat korespondensi diantara sistem yang heterogen. Pada satu tingkatan abstraksi yang tinggi, perhatian utama adalah mengidentifikasi apa yang direpresentasikan, daripada bagaimana hal itu direpresentasikan. Sebagai satu contoh dari satu sistem penunjang keputusan yang berdasarkan pada integrasi sumber sumber informasi

yang beragam, yaitu satu aplikasi bagi para job seeker atau job hunter. Seorang pengguna sistem mungkin saja tertarik pada instruksi untuk dapat menemukan informasi pekerjaan yang diinginkan dan sesuai dengan latarbelakang pendidikan dan ketrampilannya. Dia samasekali tidak tertarik pada bagaimana informasi itu disimpan atau model data apa yang sedang digunakan, tetapi pada nilai dan arti dari informasi itu sendiri. Pengguna ingin mengetahui Perusahaan mana saja yang menyediakan informasi yang diinginkan? Lebih dari itu, pengguna menginginkan mengakses informasi yang tersimpan dalam sistem yang terkait seperti informasi serupa atau yang mempunyai kemiripan dengan yang diinginkan.

## 1.1 Latar Belakang

Saat ini, fokus dari penelitian pada sistem informasi yang terpadu telah bergeser ke definisi, arsitektur dan tool yang memperbolehkan pengelolaan yang efektif dari berbagi data dalam lingkungan terdistribusi yang heterogen. Problem utama adalah volume data yang sangat besar dengan format yang juga berbeda-beda, dokumen yang semi terstruktur (XML, RDF dll.) dan spatially referenced data. Kebutuhan untuk berbagi data, bermula dari (1) perkembangan yang sangat dasyat dari sumber informasi berbasis web dan kemampuan untuk menginterkoneksi antar sumber informasi, (2) meningkatnya sumber data atau informasi yang semakin otonom, dan (3) meningkatnya biaya untuk mendapatkan data dalam format non tradisional yang semakin rumit. Sebagai contoh, informasi elektronik administrasi pemerintahan, seperti data penduduk pada dinas sosial di tingkat kabupaten akan dikumpulkan pada periode tertentu yang akan memakan biaya relatif mahal, tetapi jika ada dinas lain pada level yang sama atau berbeda akan membutuhkan data penduduk, maka dinas tersebut akan melakukan pengumpulan ulang dan tidak menggunakan data yang telah ada dari dinas yang lain.

Diilhami oleh keberhasilan sebelumnya dan kepopuleran dari sistem berbagi data file dan multimedia, seperti Napster, Gnutella, dll., arsitektur Peer-to-Peer (P2P) semakin banyak digunakan untuk berbagi sumber daya komputer, termasuk kemampuan komputasi dan penyimpanan data. Bagaimanapun, berbagi informasi yang efektif pada lingkungan yang berbasis P2P yang terdistribusi harus berbicara tentang beberapa tantangan yang penting, termasuk penemuan sumber daya (*resource discovery*), mediasi di antara format data yang heterogen, pemodelan semantik, dan pemrosesan kueri yang berbasis peer. Secara tradisional, berbagi data dari sumber sumber yang berbe-

da adalah yang dilakukan dengan cara membuat skema terpadu yang seragam untuk mengkombinasikan isi dari sumber data yang heterogen. Pendekatan ini tidak dapat dikembangkan dengan baik ketika diaplikasikan dalam lingkungan terdistribusi yang dinamis dan mempunyai banyak kelemahan sehubungan dengan peer yang berjumlah besar. Satu alternatif solusi yang lebih sesuai bagi lingkungan terdistribusi yang berorientasi pada server adalah pendekatan mediasi semantik, yang menggunakan (1) wrapper untuk memetakan skema lokal ke model data umum dan menyediakan fungsi akses data tingkat rendah dan (2) satu mediator untuk membuat *view* terintegrasi di atas sumber-sumber lokal. Mediator mendukung penemuan sumber data yang relevan dan menyediakan berbagai layanan pemrosesan kueri.

Sistem P2P berisi peer-peer yang terorganisasi dalam arsitektur yang dinamis. Peer dapat bergabung dan meninggalkan secara otonomi dan berulang. Beberapa aplikasi jaringan P2P dapat dibedakan, dimulai dari aplikasi berbagi isi (contohnya Napster [Napster, 2003], Gnutella [Gnutella, 2003]) sampai dengan aplikasi komputasi terdistribusi (seperti Avaki [Avaki, 2001], Entropia [Entropia, 2004], SETI@home [SETI@Home, 2001]) dan platform yang mendukung pengembangan [JXTA, 2004]. Pada umumnya, terdapat dua kategori utama dari sistem P2P yang telah dikenal. Sistem P2P yang tidak terstruktur mengorganisasi peer dalam tempat/ruang jaringan. Setiap peer mengontrol dan memelihara data berbaginya. Kueri pengguna berdasar pada 1) model direktori tersentralisasi di mana satu atau lebih server digunakan untuk menyimpan dan menempatkan data dan layanan, atau 2) satu model rute kueri yang secara esensial membanjiri jaringan yang menentukan peer yang relevan yang diharapkan berisi data yang diminta.

Kebalikannya, arsitektur P2P yang terstruktur mengorganisasi data dalam satu ruang kunci yang terbagi ke dalam segmen-segmen. Sejumlah segmen dialokasikan untuk tiap peer yang berpartisipasi. Kueri pengguna adalah didasarkan pada satu *Distributed Hashing Table* (DHT) yang dibangun di atas struktur *overlay* dari peer-peer. Perpanjangan dari pendekatan mediasi semantik telah diusulkan bagi sistem interoperabilitas informasi dan lebih baru lagi bagi lingkungan yang berbasis P2P. Proyek REMINDIN [Tempich et al., ] adalah sebuah ruting kueri secara semantik pada jaringan P2P berbasis metafora sosial. Sedangkan proyek P2P lainnya adalah MINERVA [Bender, 2005/2007], yang menangani data dalam jumlah sangat besar dengan cara terdistribusi dan self-organizing. Dalam sistem seperti itu, seluruh peer adalah sama dan seluruh fungsionalitasnya dibagi antara seluruh peer sehingga tidak terdapat kegagalan titik tunggal (single point failure) dan beban secara konsisten seimbang di antara sejumlah peer. Proyek CommOnCV (Competency@ontology.cv) [Bourse et al., 2002] adalah se-

buah proyek yang bertujuan pada penanganan masalah dari e-recruitment yang mempertimbangkan pendekatan baru berbasis manajemen kompetensi, semantic web dan ontologi. Tujuan dari CommOnCV adalah menyediakan layanan job matching yang baru berbasis pada pengelolaan kompetensi. Prinsip yang ada pada CommOnCV mempunyai unsur utama mempertimbangkan CV sebagai synthetic view( diekspresikan dalam bahasa alami, dalam arti kualifikasi, pengalaman kerja dan kegiatan ekstrakurikuler) dari jaringan kompetensi yang lebih kaya. Oleh sebab prinsip tersebut, tujuan pertama dari proyek adalah mengizinkan pemakai akhir membuat seluruh kompetensi yang mendasari sumberdayanya (CV nya atau pekerjaan yang ditawarkannya) secara eksplisit. Tujuan ke dua adalah merepresentasikan secara formal kompetensi tersebut dalam rangka untuk menyediakan layanan e-recruitment yang lebih berdaya: konten dari CV dan pekerjaan yang ditawarkan harus dapat dikelola oleh komputer dalam rangka menyediakan proses pencocokan (matching) secara otomatis, di sinilah peran semantic web dan ontologi.

Dalam lingkungan P2P, mediasi semantik digunakan baik untuk berbagi berkas dan ontologi sederhana yang lebih rumit berbasis penemuan informasi. APPOINT [Tanin et al., ] adalah sebuah sistem P2P yang lebih baik, yang bertujuan untuk mengurangi beban server dalam pertukaran berkas pemetaan yang besar. DBGlobe [Pitoura et al., 2003] digunakan untuk mendukung komputasi mobile untuk data GIS dengan P2P sementara P2PGIS [Bacastow and Lewis, ] adalah satu produk yang mendukung pemetaan rapid online dengan teknologi OPUS. Sistem P2P yang berbasis semantik yang lain adalah Semantic Web P2P [Arumugam et al., ] yang menyediakan satu himpunan peralatan bagi penemuan dari ontologi yang relevan, memfasilitasi penggunaan kembali ontologi yang ada dan mempublikasikan ontologi yang dihasilkan. BUSTER [Vogele and Schlieder, ] adalah satu broker data P2P, yang menawarkan pengambilan data secara intelijen, dihubungkan dengan layanan integrasi data dan translasi semantik [Visser and Schuster, ].

## 1.2 Perumusan Penelitian

Fokus dari disertasi ini adalah pada mediasi semantik dari sumber data heterogen pada lingkungan P2P. Penelitian ini bertujuan untuk mendefinisikan suatu pendekatan yang mendukung framework yang dapat menyediakan fasilitas bagi pengguna sehingga memungkinkan untuk menghubungkan sistem informasi terdistribusi, dinamis dan beragam. Berbagi informasi adalah fitur kunci dari pendekatan sistem yang diusulkan dengan melalui interaksi diantara peer yang otonom. Sistem terdahulu yang memfasili-

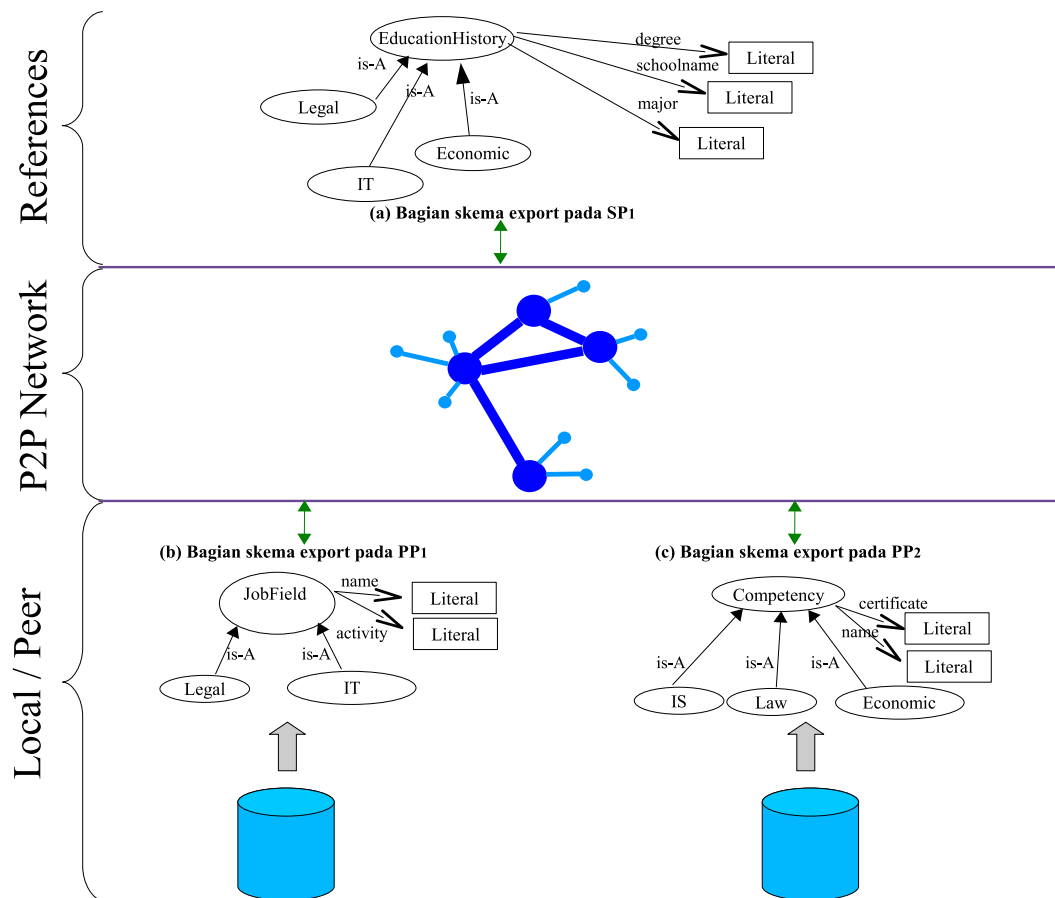
tasi berbagi informasi terutama fokus pada protokol komunikasi tingkat rendah untuk mentransfer file di antara sistem. Pada penelitian ini lebih memfokuskan pada perbedaan konsep yang terjadi antar satu peer terhadap peer lainnya.

*Untuk berbagi informasi dalam sistem ini, pertamakali penyedia data mengkonversi sekelompok data atau database ke format penerima atau model data yang umum dan kemudian mengirim hasil pada level file. Semantic interoperability setuju dengan berbagi informasi pada level abstraksi. Untuk berbagi informasi, sumber informasi yang berbeda atau peer harus membagi satu semantik umum yang generik. Setiap peer menginterpretasikan semantik ini dalam kaitan batasan dan konsep lokal. Untuk mengilustrasikan beberapa aspek penting dari berbagi informasi (information sharing) pada tingkat abstrak, maka akan dicontohkan mengenai suatu model sistem informasi pemerintahan elektronik (e-government, khususnya layanan job matching. Ini ada contoh awal.*

Dimisalkan terdapat beberapa organisasi dalam sebuah sistem P2P. Satu dari organisasi tersebut adalah suatu penyedia lapangan kerja (peer PP1) dan yang kedua adalah pencari kerja (peer PP2). Gambar 1.1 memperlihatkan fragmen dari skema ekspor dari peer PP1 dan PP2 dan bagian dari satu common ontology yang dibagi kepada peer. Peer PP1, memerlukan informasi detil dari calon pegawai baru, seperti latar belakang pendidikan, pengalaman kerja, dan keahlian. Untuk bertukar data, dua peer tersebut harus memperlihatkan (view) skema yang dimilikinya sebagai satu interpretasi dari common ontology. Satu interpretasi harus mendefinisikan korespondensi di antara konsep ontologi dan data lokal dari peer. Hal itu harus mempertimbangkan struktur, semantik dan properti dari konsep tersebut.

Riset yang diusulkan pada framework berbagi informasi adalah berbasis pada dua pondasi utama, yaitu: perbedaan versi skema/ontologi dan pemeliharaan agreement pada arsitektur P2P. Salah satu masalah adalah bagaimana mendeteksi perubahan pada ontologi referensi dan ontologi lokal, yang menyebabkan half agreement juga berubah. Sehingga isu untuk memeriksa versi dari common ontology dan juga skema lokal, membandingkan dokumen dan membandingkan dokumen OWL nya. Apabila sudah menemukan perubahan maka bagaimana melakukan penyisipan dan penghapusan dari half agreement yang lama.

Masalah utama lainnya adalah mengimplementasi pelaksanaan kueri. Hal tersebut akan mengungkapkan pencarian informasi yang mengakses satu domain. Isu pemrosesan



Gambar 1.1: Konten Peer (Bagian dari Ontologi dan Skema Ekspor)

kueri dilakukan untuk mengizinkan peer (1) untuk formulasi kueri global dalam hal konsep dari common ontology, (2) untuk mem-broadcast kueri ke peer yang relevan, dan (3) untuk mengkombinasikan respon yang paling baik ke dalam hasil akhir.

Dua fakta utama dapat dihasilkan dari diskusi di atas. Pertama, berbagi informasi didasarkan pada satu himpunan *common* dari konsep global. Arti dari konsep global adalah dispesifikasikan secara baik dan diinterpretasikan oleh sumber data dengan menggunakan proses pencocokan (*matching*) di mana konsep lokal berhubungan dengan konsep ontologi yang berkorespondensi. Mediasi semantik diantara dua peer kemudian dicapai dengan mencocokkan interpretasi tertentu yang berhubungan dari konsep ontologi global. Representasi konsep global membutuhkan pemodelan ontologi dan *tool* yang berkaitan. Ontologi secara meningkat digunakan untuk menggambarkan arti dan keterhubungan diantara konsep dari domain aplikasi. Seperti yang didefinisikan oleh Gruber dalam [Gruber, 1995], sebuah ontologi adalah suatu deskripsi eksplisit dari suatu konseptualisasi entitas dunia nyata dari satu domain aplikasi. Ontologi adalah kosakata yang disusun dari *term* dan relasinya. Beberapa bahasa telah diusulkan un-

tuk memodelkan dan menalarakan ontologi. Bahasa tersebut berjajar dari mulai bahasa alami informal sampai bahasa formal berbasiskan pada logika predikat atau konsep graf. Diantara bahasa formal, OWL yang saat ini mempunyai kepopuleran, adalah bahasa deskripsi yang bertujuan pada penggabungan semantik berbasis teori dan penyimpulan secara ontologi dan mekanisme penalaran ke dalam RDF.

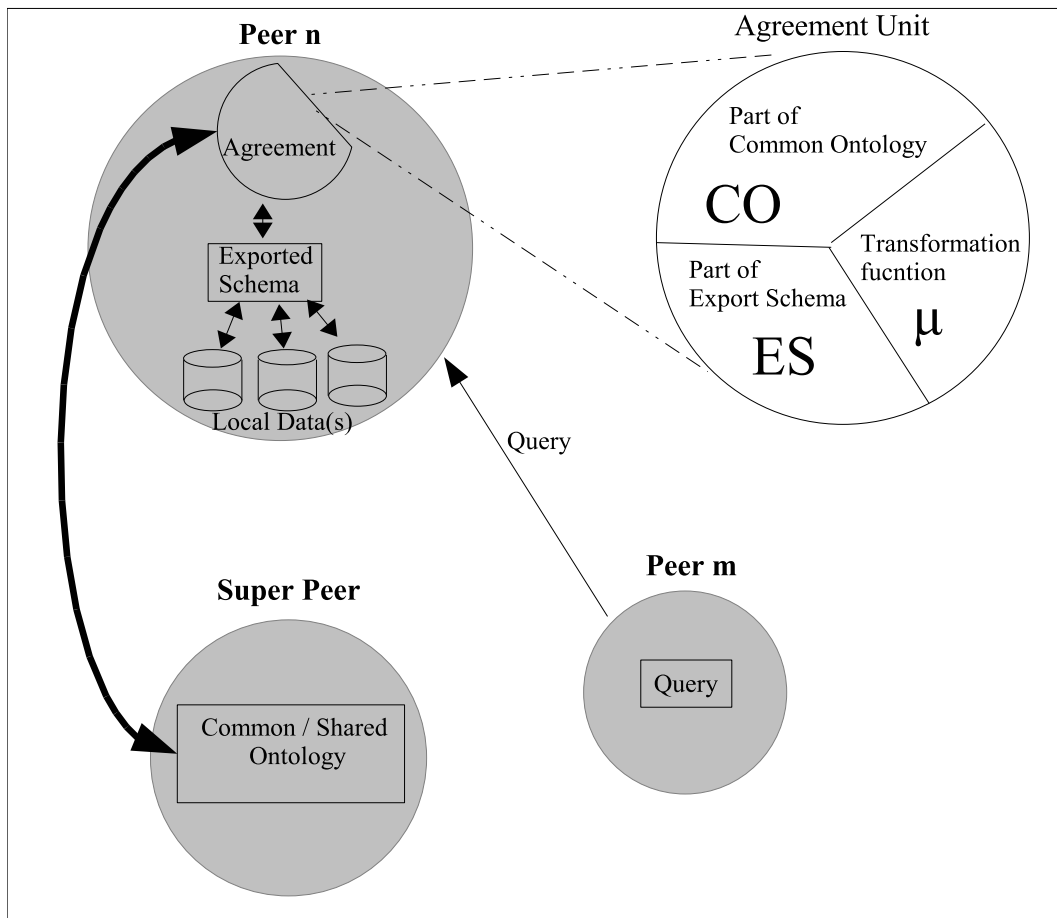
Sebagai ringkasan, riset ini fokus pada pertanyaan-pertanyaan berikut ini:

- Bagaimana mendeteksi perubahan common ontology dan local schema ?
- Bagaimana melakukan perbandingan dokumen OWL ?
- Bagaimana memberikan usulan pertimbangan agar dapat menyesuaikan agreement apabila tidak ada yang berubah, sebagian saja yang berubah dan semuanya berubah ?

### 1.3 Gambaran Umum

Tujuan utama dari penelitian ini adalah mengembangkan satu pendekatan untuk mengetahui perubahan skema lokal atau ontologi sehingga perlu melakukan pemeliharaan agreement antara suatu common ontology dan provider peer yang ada. Dalam lingkungan yang dinamis dan otonom, administrator harus secara terus menerus memonitor common ontology. Monitoring secara manual adalah time consuming dan tidak akan dapat diprediksi jumlah anggota peer dengan baik. Hal yang penting adalah mengembangkan teknik yang secara otomatis memonitor dan melakukan usulan perbaikan *agreement*.

Ide kunci dari riset ini adalah konsep dari pemeliharaan semantic peer agreement, di mana pada gambar 1.2 diperlihatkan konsep semantic agreement itu sendiri. Mediasi semantik di antara peer melibatkan pembuatan unit agreement untuk menjembatani kesenjangan semantik di antara isi yang terdapat dalam peer. Hal ini membutuhkan definisi dan komputasi dari alat ukur kesamaan semantik diantara konsep-konsep yang digunakan oleh peer yang berbeda. Untuk membuat unit agreement dan juga half agreement yang telah dihasilkan selalu dalam keadaan terkini, maka harus dilakukan pemeliharaan. Pemeliharaan agreement dapat dilakukan dengan cara memeriksa versi dari common ontology dan skema lokal dan kemudian melakukan perbandingan baik untuk dokumen juga dokumen OWL. Selanjutnya hal yang perlu dilakukan adalah mendapatkan informasi yang berasal dari domain yang ada. Pendekatan yang diusulkan meliputi isu di bidang (1) arsitektur P2P (2) Mediasi semantik (3) Pemrosesan kueri.



Gambar 1.2: Komponen dari *Peer Agreement Approach*

### 1.3.1 Arsitektur

Dalam pendekatan agreement, peer bertukar informasi melalui publikasi kontennya dan penubmitan kueri untuk mengambil konten dari peer yang lain. Untuk melakukan hal tersebut, peer membagi common ontology. Model RDF/OWL digunakan untuk merepresentasikan konten dari peer. Akan digunakan definisi yang dilakukan oleh [Wicaksana, 2006] yaitu arsitektur P2P hybrid 2 (dua) level, di mana satu peer dapat beraksi sebagai super peer atau pun sebagai simple peer. Sebagai simple peer, peer tersebut mengekspor dan membagi kontennya dengan yang lain. Setiap simple peer mengandung hal berikut ini: (1) satu skema ekspor yang mendefinisikan satu view di atas data berbagi yang dapat diakses ke peer lain dan (2) informasi agreement untuk membuat link di antara skema ekspor dan common ontology.

Sebagai super peer, peer tersebut mendaftarkan peer aktif dan mendefinisikan metadata yang digunakan untuk menggambarkan konteks mediasi semantik. Super peer meme-

lihara ontologi global dan menyediakan taksonomi konsep bagi domain umum. Ketika sejumlah besar peer bergabung dan meninggalkan arsitektur, super peer mungkin saja dapat kelebihan beban dan menjadi titik tunggal kegagalan (single point failure). Untuk menghindari ini, sejumlah tugas yang diberikan kepada super peer harus dikurangi atau multiple super peer dapat digunakan untuk membagi tugas pengelolaan.

### 1.3.2 Mediasi Semantik

Tiga hal yang merupakan hal penting dalam pendekatan mediasi semantik, ontologi, semantic similarity, dan agreement.

**Ontologi:** ontologi digunakan sebagai titik acuan oleh peer untuk mempublikasikan data dan meminta dari peer yang lain. Sebuah ontologi menyediakan spesifikasi untuk meng-capture semantik dari konsep yang dibagi oleh aplikasi dalam domain yang telah diberikan. Ontologi adalah kosa kata yang dibangun dari term dan relasi diantaranya. Beberapa bahasa dan format dapat digunakan dalam menggambarkan ontologi, termasuk bahasa alami dan metode formal seperti RDF, OWL dan bahasa deskripsi lainnya. Di sini sudah diasumsikan bahwa ontologi yang ada digambarkan oleh ahli domain dengan menggunakan bahasa representasi seperti RDF, dan lainnya. Sekali ontologi sudah diberikan, konsep yang dispesifikasikan digunakan sebagai elemen dasar di mana perbandingan semantik konten dari peer dilaksanakan. Sebagai ringkasan, ontologi menyediakan pengguna peer dengan kosa kata, kumpulan term dan relasi, untuk mengkonversi dan bertukar informasinya.

**Semantic Similarity:** Salah satu cara untuk membandingkan dua konsep adalah mengkalkulasi semantic similarity-nya. Semantic similarity dapat dilakukan dengan pertamakali mencocokkan label yang berasosiasi dengan konsep untuk menentukan konsep yang berhubungan. Pencocokan label (label matching) terdiri dari dua langkah utama. Langkah pertama adalah pemrosesan bahasa, digunakan untuk mentransform atau memecah label menjadi kata yang terpisah. Sebagai contoh, langkah ini dapat digunakan untuk memanjangkan singkatan dan mengganti akronim. Langkah ke dua adalah analisis linguistik yang digunakan untuk menemukan hubungan diantara satu label atau bagian dari label dan kata lain. Biasanya, hubungan dapat berupa sinonim, antonim dan lainnya. Langkah ini bersandar pada satu thesaurus seperti WordNet, dan lainnya. Pencocokan label diaplikasikan pada level konsep paling atas untuk menghitung ukuran similaritas. Hasil dari proses pencocokan ini dapat disesuaikan / diperhalus dengan memikirkan baik struktur internal dan struktur eksternal dari konsep yang direpresen-

tasikan oleh label. Struktur internal dari konsep mengandung atribut yang terhubung langsung sementara struktur eksternal adalah memikirkan posisi konsep dalam hirarki.

**Agreement:** *Semantic similarity* adalah langkah yang kritis dalam menentukan pemetaan di antara konsep dari simple peer dan konsep dari common ontology yang dikelola oleh super peer. Agreement meliputi satu atau lebih unit deskripsi semantik, yang dinamakan sebagai unit agreement atau half agreement [Wicaksana, 2006], yang digunakan untuk menyimpan pencocokan yang berkoresponden diantara konsep. Lebih tepat lagi, sebuah unit agreement mendefinisikan interpretasi atau view dari satu atau lebih konsep ontologi, sehingga mengadaptasi semantik dari konsep berbagi global ke view lokal dan batasan dari peer. Dalam bentuknya yang sederhana, unit agreement mendefinisikan pemetaan (mapping) antara (,) di mana adalah konsep common ontology dan adalah konsep skema ekspor. Sebuah unit agreement atau half agreement terdiri dari tiga komponen: konsep ontologi, konsep skema ekspor yang berelasi dan fungsi transformasi yang digunakan untuk memetakannya. Metadata disimpan dari pembuatan unit agreement yang digunakan dalam proses kueri untuk menemukan peer yang relevan. Ketika dua simple peer perlu untuk membagi informasi, simple peer tersebut pertama kali mendefinisikan unit agreement yang sesuai. Kemudian, simple peer tersebut mengkombinasikan unit agreement menjadi agreement penuh untuk memulai pemetaan semantik langsung antara konsep-konsepnya.

### 1.3.3 Pemrosesan Kueri

Langkah berbeda dilakukan oleh peer ketika peer-peer itu submit atau menerima kueri. Pemrosesan kueri diorganisasi dalam dua fase. Fase pertama dialokasikan untuk definisi dan penyimpanan common ontology dan konteks metadata dalam satu atau lebih super peer, yang mengkoordinasi pertukaran data dan kegiatan berbagi. Fase ke dua, peer submit dan menerima kueri melalui dua tipe pesan: Publish dan Request. Ketika peer membuat unit agreement, simple peer mengirim sebuah pesan Publish ke super peer. Pada saat menerima pesan Publish, super peer meng-update konteks metadatanya yang digunakan dalam proses penemuan. Untuk mengakses data remote, pertamakali, simple peer mengirim pesan Request ke super peer yang membalas dengan sebuah daftar dari peer kandidat. Kemudian, peer yang meminta berusaha untuk menggabungkan/menghubungkan konsepnya ke konsep dari satu atau lebih peer pada daftar kiriman kandidat dari peer. Proses pengikatan didasarkan pada pemetaan ontologi ditentukan dengan pesan Publish dalam langkah publish di atas.

### 1.3.4 Kueri

**Persiapan Eksperimen:** Sebuah prototipe arsitektur yang telah dikembangkan oleh Wicaksana [Wicaksana, 2006] dan contoh implementasi yang telah didesain digunakan untuk memvalidasi usulan pertimbangan pemeliharaan semantik *agreement*.

**Hasil Eksperimen :** Dalam skenario evaluasi, terdapat beberapa peer, yang mempunyai konsep berbeda. Konsep menggunakan banyak tipe klasifikasi dari sistem informasi pemerintahan elektronik untuk layanan *job matching* seperti yang dijadikan diskusi dalam contoh. *Common ontology* berbasiskan HR-XML. *Local ontology* dikembangkan dari kombinasi beberapa sumber, seperti beberapa penyedia jasa layanan penyedia lapangan pekerjaan (*job provider*), dan beberapa pencari kerja (*job seeker*).

## 1.4 Kontribusi

Hasil utama dari penelitian ini adalah spesifikasi dari suatu framework berbasis pada satu pendekatan semantic agreement untuk lingkungan P2P. Framework ini didasarkan atas arsitektur P2P model hybrid dua level yang telah dikembangkan oleh [Wicaksana, 2006] yang terdiri dari dua tipe peer: (1) super peer yang digunakan untuk me-register dan mengelola peer lainnya, dan (2) simple peer yang isinya diekspor dan dibagi dengan peer yang lain. Penelitian ini mengembangkan suatu model untuk mengusulkan pemeliharaan semantic agreement pada lingkungan P2P, sehingga pendekatan sebelumnya yang cenderung statis akan menjadi dinamis karena pendekatan ini akan dapat mendeteksi perubahan yang terjadi dalam *agreement* dan melakukan usulan penyesuaian terhadapnya.

Akan dikembangkan prototipe simulasi untuk mengevaluasi dan memvalidasi beberapa konsep dan ide yang didiskusikan dalam disertasi ini. Sebagai ringkasan, kontribusi utama dari disertasi ini adalah : mengembangkan satu model usulan pertimbangan apakah pendekatan mediasi yang menggunakan semantic agreement yang telah dikembangkan Wicaksana dapat terpelihara, Sehingga didapatkan usulan pertimbangan apakah akan menggunakan pendekatan semantic agreement yang kompleks atau yang sederhana.

## 1.5 Organisasi Desertasi

Desertasi ini disusun sebagai berikut. Bab 2 menyajikan penelusuran literatur pada dua bidang penelitian utama yang digunakan dalam desertasi ini: mediasi semantik khususnya pendekatan semantik agreement dan arsitektur peer-to-peer. Bab 3 meringkas aspek yang penting dalam pendekatan semantik agreement. Pada bab 4 akan dikenalkan isu pertimbangan pemeliharaan semantik agreement dalam lingkungan P2P sehingga akan mendapat gambaran untuk menghasilkan usulan pertimbangan apakah akan menggunakan algoritma dalam pendekatan semantic agreement yang kompleks atau yang lebih sederhana. Sedangkan bab 5 mendiskusikan hasil dari simulasi yang digunakan untuk mengevaluasi usulan pertimbangan pemeliharaan semantik agreement. Dan akhirnya pada bab 6 menyajikan kesimpulan dan penelitian yang dapat dilanjutkan dari penelitian dalam desertasi ini.

## Bab II

# TINJAUAN PUSTAKA

Pada bab ini akan melihat dua domain penelitian utama dengan menitik baratkan pada pertimbangan dalam pemeliharaan dan pendekatan mediasi semantik, yaitu: semantic agreement dan lingkungan Peer-to-Peer (P2P). Dalam bahasan tinjauan pustaka akan mengacu kepada semantic agreement, pencocokan skema, pemetaan ontologi dan manajemen perubahan dalam ontologi. Sedangkan dalam bahasan P2P akan melihat definisi dan karakteristik P2P, P2P discovery, teknik pencarian informasi dan pengindeks-an dalam P2P.

### 2.1 Semantic Agreement

Istilah semantic agreement yang berkaitan dengan interoperabilitas semantik maupun *semantic integration*, muncul pertamakali pada [Mackay, 1999] dalam suatu penelitian mengenai *semantic integration* dari suatu model dari lingkungan hidup untuk aplikasi GIS dan pengambilan keputusan. Dalam perkembangannya istilah tersebut ternyata jarang sekali muncul pada saat pencarian dengan menggunakan beberapa macam mesin pencari, seperti Google, Yahoo, Wikipedia dan Answers. Sampai kemudian istilah *semantic agreement* diadaptasi oleh Wicaksana dalam penelitiannya mengenai pengembangan pendekatan semantic agreement berbasis P2P untuk interoperabilitas sistem informasi [Wicaksana, 2006]. Untuk mengatasi perbedaan semantik di antara konsep peer, maka *agreement* dibuat untuk merepresentasikan hasil dari pencocokan (matching) atau pemetaan (mapping) satu konsep ke konsep dari peer yang lain.

Pembuatan pemetaan semantik di antara sumber informasi yang berbeda adalah salah satu titik yang kritis dalama pendekatan interoperabilitas. Banyak penelitian yang

ada saat ini adalah berbasis pada ide bahwa pemetaan (*mapping*) dapat dengan mudah dibuat oleh perancang yang ahli ketika skema dari sumber informasi yang berbeda dikombinasikan dan diintegrasikan. Seperti kebalikannya, identifikasi dari keterhubungan semantik di antara sumber informasi yang berbeda adalah satu masalah yang sulit walaupun pada sistem terpusat jika akan di bawa ketataran mesin dapat memahaminya.

Konflik semantik timbul ketika dua sistem tidak menggunakan interpretasi yang sama terhadap suatu informasi. Bentuk paling sederhana dari ketidaksetujuan dalam menginterpretasikan informasi adalah homonim (menggunakan kata yang sama dengan arti yang berbeda), dan sinonim (menggunakan kata berbeda dengan arti yang sama). Dalam kasus ini, semantik dari informasi harus dipertimbangkan dalam rangka memutuskan bagaimana item informasi yang berbeda berhubungan satu dengan yang lainnya. Yaser [Bishr, 1997] telah membagi keragaman skematik ke dalam beberapa kelompok:

- Berbeda dalam kelas seperti sinonim, homonim, berbeda dalam atribut kelas, metode dan batasan integritas.
- Berbeda dalam atribut, seperti domain, unit, nilai default dan tipe data
- Berbeda dalam hirarki, seperti kelas, atribut, tingkat generalisasi dan agregasi.
- Berbeda dalam hirarki, seperti kelas, atribut, tingkat generalisasi dan agregasi.

Untuk mengatasi perbedaan semantik di antara konten dari peer, maka agreement dibuat untuk merepresentasikan hasil dari pencocokan atau pemetaan konsep dari satu peer ke konsep dari peer yang lain. Seperti yang sudah didiskusikan di atas, pencocokan konsep adalah kombinasi dari (1) analisis linguistik pada label konsep, (2) pencocokan label yang berdasarkan pada definisi dari nilai similaritas menggunakan struktur (internal dan eksternal) dari konsep dan (3) identifikasi dari relasi semantik di antara konsep. Dalam langkah yang berbeda dari pencocokan semantik, sebuah konsep dipandang sebagai hirarki dari label terminologikal yang berisi informasi struktur dan semantik. Pendekatan agreement dalam Wicaksana berdasarkan beberapa asumsi yang meliputi fakta bahwa bahasa yang sama digunakan untuk merepresentasikan skema ekspor dan ontologi yang terdapat dalam agreement context dan konteks komunitas secara bersamaan, di mana label dari konsep tersebut mengandung beberapa informasi semantik dan pada akhirnya tidak ada individual ataupun instant yang berasosiasi dengan common ontology.

### 2.1.1 Pencocokan Skema dan Pemetaan Ontologi

Pencocokan (matching) adalah satu dari pekerjaan yang penting dalam interoperabilitas informasi [Paoli and Disco, 1005]. Secara umum, proses pencocokan membutuhkan dua skema sebagai masukan yang mengandung kelas, properti, aturan dan lain lain. Hasil dari pencocokan menentukan keluaran dari relasi antara elemen. Yang merupakan motivasi mengapa masalah pencocokan sangat penting adalah sebagian besar sumber informasi mempunyai model yang berbeda untuk merepresentasikan konten dari informasi. Sebagai contoh, satu sumber mempunyai kelas *Computer\_server*, sumber yang lain mempunyai kelas *Computer*. Ketika sebuah permintaan dikirim ke dua sumber daya tadi, maka sebuah pemetaan diperlukan untuk memperhitungkan / mempertimbangkan relasi antara *Computer\_server* dan *Computer*.

Masalah pencocokan telah dipelajari dalam basis data federasi yang ada saat ini dan bidang integrasi data lainnya. Dalam basis data federasi, pemetaan ditegakkan antara skema dari basis data individual. Pemetaan bilateral dibuat untuk setiap pasang basis data, menghasilkan n2 pemetaan. Metode yang lain adalah mendefinisikan sebuah skema global untuk memasukan keseluruhannya dan membuat pemetaan dari skema global ke skema lokal, yaitu pendekatan *global as view* (GAV) dan *local as view* (LAV) [Cali et al., ].

Tujuan utama dari pemetaan ontologi adalah untuk mengembangkan pengertian antara kelompok, representasi formal untuk berbagi konseptualisasi (seperti, tabel, elemen XML, kelas, properti, aturan dan predikat) dan relasinya. Hasil dari konseptualisasi adalah sebuah himpunan dari istilah (term) dan keterhubungan (relasi) antar istilah (ekuivalen, subsumption). Tapi sayang sekali, tidak mudah untuk membuat setiap orang setuju pada ontologi yang sama pada sebuah domain. [Kalfoglou and Schorlemmer, ]. Saat ini, terdapat sejumlah besar penelitian pada pencocokan skema dan pemetaan ontologi. Perbedaan yang utama antara pencocokan skema dan pemetaan ontologi adalah [Sinir, ]:

### 2.1.2 Manajemen Perubahan dalam Ontologi

Ontologi, adalah spesifikasi eksplisit dari konsep [Gruber, 1995], dapat digunakan dalam konteks untuk menyediakan Sistem Ekstraksi Informasi (IES) dengan definisi dari informasi yang relevan dalam suatu cara yang formal dan dapat dimengeti komputer. Ontologi mendapatkan perhatian lebih pada akhir akhir ini, seperti pemikiran sebagi-

an besar peneliti bahwa ontologi dapat melayani sebagai backbone dari Semantic Web [Yildiz, 2006]. Semantic Web adalah sebuah perluasan dari Web yang saat ini ada dengan semantik menyertai data, yang dapat membuat lebih mudah untuk manusia dan mesin untuk menemukan informasi [Berners-Lee and Connolly, 1998]. Tetapi area aplikasi dari ontologi tidak terbatas digunakan dalam Semantic Web. Ontologi adalah objek yang berhubungan dengan pengetahuan dan dapat digunakan dalam area aplikasi apa saja di mana domain ketertarikan (domain of interest) harus dikonseptualisasikan. Bagaimanapun juga, penggunaan ontologi dalam Sistem Informasi (IS) membutuhkan juga sebuah manajemen yang akurat. Isu yang paling penting dalam konteks tersebut adalah tetap memelihara ontologi tetap up-to-date mengikuti perubahan dalam domainnya yang digunakan, karena sebagian besar domain mempunyai sifat yang dinamis [Yildiz, 2006]

Dunia ini secara teratur berubah, hal tersebut mempengaruhi yang terdapat di dalamnya. Pada sebagian aplikasi, ontologi tidaklah statis. Selain itu, ontologi harus beradaptasi dengan perubahan dari domain aplikasi dan perluasan area. Sehingga evolusi ontologi adalah salah satu aspek dalam pemeliharaan ontologi [Borden, 2007]. Noy dan Klein [Klein and Bernstein, 2003] bergumentasi bahwa evolusi ontologi sangat dekat dengan evolusi skema dalam basisdata, tetapi evolusi ontologi mempunyai beberapa kekhususan. Terutama, evolusi ontologi adalah berbeda semantik dan berbeda paradigma. Klein et.al [Klein and Bernstein, 2003] membedakan perubahan konseptual (cara di mana sebuah domain dipahami) dari perubahan penjelasan (cara bagaimana konsep dispesifikasikan). Dalam [Noy, 2003], perubahan ontologi terlihat seperti urutan dari operasi update secara individual seperti sebuah log file dari suatu sistem basis data. Pada paper tersebut didiskusikan transformasi paling minimal di antara dua status ontologi, misalnya bagaimana berpindah dari satu status ke status yang lain dengan himpunan terkecil dari update secara individu dan bagaimana mengkonstruksi operator update yang rumit dari urutan update secara individual. Operasi update tersebut dapat secara sendirinya dapat diorganisasi sebagai sebuah ontologi dan ditawarkan kepada pengguna dalam sebuah menu [Cali et al., ].

Ontologi seringkali dipandang sebagai blok bagi semantic web, sehingga ontologi menyediakan sejumlah pengetahuan yang reusable mengenai domain yang spesifik. Bagaimanapun juga, sejumlah pengetahuan yang reusable tersebut seringkali tidak statis, tetapi berkembang sepanjang waktu. Domain berubah, adaptasi pada tugas-tugas yang berbeda, atau perubahan dalam konseptualisasi, membutuhkan modifikasi dari ontologi.

Manajemen perubahan dalam ontologi dapat mempunyai beberapa bentuk: [Yildiz, 2006]

- ontology modification: merubah ontologi tanpa mengacuhkan mengenai konsistensinya
- ontology versioning: membangun dan mengelola versi-versi yang berbeda dari sebuah ontologi dan menyediakan akses untuk versi-versi tersebut.
- ontology evolution: mengubah ontologi dan membuatnya konsisten.

Terlihat jelas bahwa istilah *ontology versioning* dan *ontology evolution* telah diadopsi oleh komunitas perancangan ontologi, seperti banyak peneliti berpikir bahwa bidang ini serupa dengan bidang *evolution* dan *versioning* dalam komunitas basis data. Walaupun, banyak peneliti lainnya menyatakan bahwa terdapat perbedaan mendasar antara kedua bidang tersebut dan tidak lah mungkin untuk membedakannya antara *ontology versioning* dan *evolution* [Klein and Noy, 2003], istilah yang ada dan masih tetap digunakan secara luas.

**Merepresentasikan Perubahan Ontologi** Salah satu isu dalam konteks manajemen perubahan adalah representasi yang layak dari perubahan. Cara paling mudah dan paling sederhana adalah selalu menjaga perubahan dalam bentuk daftar perubahan (*change log*) yang berisi urutan yang pasti, akurat dan tepat dari perubahan yang diaplikasikan pada sebuah ontologi. Walaupun mudah, hal itu bisa saja menjadi sebuah masalah untuk membuat *change log* tersebut tersedia. Bagi sebuah komunitas terdistribusi dari pengembang ontologi dan atau para pengguna. Bagaimana pun, hal itu dapat menjadi berguna bagi pengembangan ontologi lokal, oleh karena itu, *change log* didukung oleh beberapa tool *ontology-editing*, seperti Protege atau OntoEdit

Menurut Klein [Klein and Bernstein, 2003], sebuah spesifikasi perubahan yang komprehensif sebaiknya mengandung paling sedikit informasi berikut ini:

- sebuah spesifikasi operasional dari sebuah perubahan,
- relasi konseptual antara versi lama dan baru dari konsep yang berubah,
- meta data mengenai perubahan,
- relasi evolusi antara konsep dalam versi lama dan baru,
- dan informasi tentang tugas atau konsekuensi perubahan dari spesifik domain.

Selain change log, Klein dan Noy [Klein and Noy, 2003] menyebutkan tiga kemungkinan lainnya untuk mengakses dan merepresentasikan perubahan antara dua versi ontologi: melakukan sebuah structural diff, merepresentasikan perbedaan dalam bentuk perubahan konseptual atau dalam bentuk himpunan transformasi.

Cara yang pertama, adalah melakukan *structural diff* sebagaimana dijelaskan dalam [Noy and Musen, ]. Mereka menggambarkan sebuah perbandingan dalaman bidang software code versioning, di mana kode juga sering berubah dan perbedaan antara dua versi dapat diakses dengan menggunakan sebuah proses yang dinamakan diff; yang menghasilkan sebuah daftar dari barisan yang hal itu berbeda dalam dua versi. Pengarangnya menyatakan, bagaimanapun, bahwa pendekatan ini tidak dapat diturunkan untuk membandingkan dua versi ontologi, karena bentuk dari representasi dan bentuk dari ontologi yang dihasilkan sangatlah berbeda dengan software code. Maka, akan lebih memungkinkan apabila dua ontologi adalah identik dalam bentuk konseptualisasinya, tetapi sangat berbeda dalam bentuk representasi internalnya. Sehingga, Noy dan Klein [Klein and Noy, 2003] mengusulkan sebuah algoritma yang dinamakan PROMPTDIFF (algoritma ini tersedia sebagai plugin pada lingkungan Protege 2000 ontology-editing) untuk membandingkan dua ontologi dengan mengacu kepada strukturnya dan bukan representasi teksnya. Untuk setiap struktur ontologikal dalam versi yang lama dari sebuah ontologi, akan mencari kemungkinan struktur yang saling terkait (koresponding) dalam versi yang baru. Jika terdapat beberapa struktur di mana tidak ada pasangan langsung yang dapat ditemukan, beberapa heuristic akan diaplikasikan untuk mencari kemungkinan cocok. Kemudian berusaha untuk menggabungkan hasil tersebut dengan menggunakan sebuah algoritma fixedpoint. PROMPTDIFF dapat mencapai rata rata nilai recall 96% dan rata rata nilai precision 93%.

Cara kedua, adalah merepresentasikan perbedaan antara ontologi dalam bentuk perubahan konseptual. Perubahan seperti itu menspesifikasikan relasi konseptual antara struktur ontologikal dalam versi yang lama dan versi baru. Sebagai contoh, sebuah perubahan konseptual dapat dinyatakan bahwa sebuah konsep A adalah sebuah subkonsep dari B dalam versi lama sebelum berpindah ke tempatnya dalam versi baru.

Cara ketiga, adalah merepresentasikan perubahan antara ontologi dalam bentuk himpunan transformasi, di mana mengandung operasi perubahan yang menspesifikasikan bagaimana (misalnya mengaplikasikan yang berubah) versi lama dari ontologi dapat ditransformasi menjadi versi baru. Himpunan transformasi berbeda dari change log yang sederhana yang hanya mengandung operasi yang diperlukan untuk mencapai ver-

si (perubahan) yang dimaksud. Lebih jauh, himpunan transformasi dapat saja tidak mengandung perubahan dalam urutan yang sama seperti kenyataannya mereka diaplikasikan. Sebagai contoh, menambah (*add*) komponen baru dapat dikelompokkan bersama, karena hal itu tidak mempengaruhi ontologi secara keseluruhan seperti operasi penghapusan (*delete*).

Dalam mendeteksi perubahan yang ada dalam suatu ontologi dapat dilakukan dengan berbagai metode. Beberapa metode yang telah ditemukan dan dikembangkan adalah, pendekatan berbasis version log yang dikemukakan oleh Peter Plessers dalam [Plessers et al., 2007], pendekatan deteksi perubahan yang dikembangkannya diklaim sebagai hal baru karena pendekatan tersebut memungkinkan pengguna yang berbeda untuk mendefinisikan himpunan yang berbeda dari definisi perubahan yang direpresentasikan perubahan yang diminati oleh mereka, dan memungkinkan menspesifikasikannya pada tingkatan abstrak yang paling sesuai dengannya. Definisi perubahan ini kemudian digunakan untuk mendeteksi bagian yang mana dari perubahan tersebut terjadi. Dengan cara seperti itu, setiap pengguna memperoleh sejenis view personal pada perubahan yang diaplikasikan pada ontology. Hal itu akan menyediakan dukungan yang lebih baik untuk memahami evolusi ontologi. Untuk merealisasikannya, pendekatan deteksi perubahan berbasiskan dua elemen kunci: version log dan Change Definition Language (CDL). Tujuan dari version log adalah untuk menyimpan jejak dari versi-versi yang berbeda dari seluruh konsep yang pernah dibuat di ontologi. Ketika sebuah konsep ditambah (*added*), modifikasi (*modified*), atau dihapus (*removed*), version log di-update untuk merefleksikan status yang baru. Adalah penting untuk mencatat bahwa kita menyimpan versi dari konsep individual dan bukan versi dari ontologi itu sendiri. Tentu saja, sebuah versi ontologi dapat dibangun kembali (*reconstruct*) dari versi konsep. CDL berbasis pada temporal logic dan memungkinkan pengguna untuk mendefinisikan perubahan yang diminatinya. Sebagai tambahan, definisi perubahan yang dispesifikasikan dengan cara bahasa tersebut digunakan untuk mendeteksi yang mana dari perubahan tersebut terjadi. Hal itu dilakukan dengan mempertimbangkan definisi perubahan sebagai temporal queries pada version log.

Pendekatan deteksi [Stojanovic, 2004] perubahan ontologi lain terdapat dalam yang mendefinisikan evolusi ontologi sebagai adaptasi tepat waktu dari sebuah ontologi pada perubahan yang timbul dan propagasi secara konsisten dari perubahan tersebut pada object yang bergantung. Dalam [Stojanovic et al., 2002], pengarang mengidentifikasi sebuah kemungkinan proses evolusi. Untuk merepresentasikan perubahan, mereka memperkenalkan dalam tiga level abstraksi untuk perubahan ontologi untuk bahasa

KAON [Maedche et al., ]. Mereka membedakan: perubahan mendasar (modifikasi untuk satu entitas ontologi), perubahan komposit (modifikasi pada tetangga langsung dari sebuah entitas ontologi) dan perubahan kompleks (modifikasi pada sebuah himpunan sembarang dari entitas ontologi). Framework itu telah dikembangkan lebih lanjut oleh Haase [Haase and Stojanovic, 2005] termasuk dukungan bagi bahasa OWL.

Sebuah framework evolusi ontologi bagi bahasa OWL juga dikembangkan oleh Klein et al [Klein and Noy, 2003]. Serupa dengan [Stojanovic, 2004], dia juga memberikan taksonomi serupa bagi bahasa OWL untuk mendefinisikan kedua operasi perubahan, baik operasi dasar dan operasi kompleks. Operasi dasar perubahan pada entitas ontologi tunggal sedangkan operasi perubahan kompleks adalah mekanisme untuk mengelompokkan operasi dasar bersama-sama untuk membentuk sebuah unit logikal. Lebih jauh, dalam [Klein and Noy, 2003], dia mengindikasikan kegunaan dari pendekatan deteksi perubahan komposit. Pengarang memperkenalkan mekanisme deteksi berbasis pada aturan dan heuristic untuk mendeteksi perubahan komposit di antara dua versi ontologi (Vold and Vnew).

Penelitian lain yang masuk kategori evolusi ontologi, khususnya dalam area deteksi perubahan adalah pada [Huang and Stuckenschmid, 2005], dia mengembangkan sebuah framework untuk reasoning dengan ontologi multi-versi. Pada pendekatan tersebut menggunakan temporal logic sebagai fondasi dari reasoning multi-versi.

### **Ontology versioning**

Definisi dari ontology versioning adalah kemampuan untuk menangani perubahan dalam ontologi dengan membuat dan mengelola variasi yang berbeda dari ontologi tersebut [Klein et al., 2002]. Tujuannya adalah mengeksplorasi sebanyak mungkin pengetahuan [Vrije Universiteit, 2004]. Ontology versioning harus dapat memungkinkan mengelola versi berbeda dari ontologi yang sama pada satu waktu. Fungsionalitas ini sangat penting dalam suatu keadaan di mana pengembang ataupun pengguna dari ontologi akan mengakses sebuah ontologi dengan cara terdistribusi. Mempertimbangkan bahwa satu dari benefit utama dari ontologi adalah re-use dan interoperabilitas yang mereka sediakan, situasi tersebut melebihi sebuah teoritikal.

Saat ini, belum ada mekanisme versioning yang mutakhir yang tersedia. Yang terjadi adalah, ontologi berubah dan versi lama hilang untuk selamanya, karena hanya yang versi terakhir yang dapat di akses. Kadang-kadang, versi lama dan baru dari ontologi diarsipkan, tetapi tidak ada mekanisme yang menyediakan penekanan / menggarisba-

wahi perbedaan antara versi. Yang pertamakali mencoba untuk membahas masalah ini adalah Heflin dan Hendler [Heflin et al., 2003] dengan memperkenalkan Simple HTML Ontology Extensions (SHOE) sebagai perluasan dari HTML untuk merepresentasikan pengetahuan berbasis ontologi menggunakan tag. Satu fasilitas yang penting adalah bahwa SHOE memungkinkan pengembang ontologi untuk menyatakan apakah sebuah versi adalah kompatibel secara berlawanan dengan versi yang lama atau tidak. Dalam sebuah bidang aplikasi terdistribusi, di mana banyak kelompok-kelompok berinteraksi (misalnya, aplikasi, agent, dll) menggunakan ontologi yang sama, informasi ini sangat penting, karena informasi tersebut menentukan apakah mereka dapat melanjutkan pekerjaan mereka sebagaimana biasanya atau mereka harus meng-update versinya dalam rangka untuk memelihara interaksi agreed-upon.

Bagaimanapun, trend Semantic Web saat ini perlu membuat pendekatan lebih mutakhir untuk ontology versioning. Dalam [Klein et al., 2002] mengenakan kebutuhan berikut ini pada ontology versioning framework:

- Identifikasi: definisi yang dimaksud dari komponen ontologikal harus dibuat jelas sebelumnya, karena hal itu merepresentasikan prasyarat untuk spesifikasi perubahan.
- Spesifikasi Perubahan: kemungkinan perubahan harus dispesifikasikan sesuai dengan identifikasi dari komponen ontologikal. Karena, paradigma representasi yang berbeda menyediakan komponen yang berbeda, spesifikasi perubahannya pun juga akan berbeda.
- Evolusi transparan: harus secara jelas aksi apa yang harus dilakukan ketika perubahan tertentu terjadi. Untuk tujuan tersebut, spesifikasi perubahan akan digunakan untuk mentranslasi dan menghubungkan versi yang berbeda dari komponen ontologikal.
- Kepedulian tugas: karena, terdapat dimensi yang berbeda sampai kompatibilitas (misalnya, preservasi dari data instant, preservasi dari hasil kueri, preservasi dari konsekuensi, dll.), sebuah framework harus peduli tugas yang kongkret dalam rangka menyediakan transformasi yang tepat di antara versi-versi.
- Dukungan bagi perubahan yang tidak terlacak: Sering kali kasusnya, ketika tidak terdapat jejak dari perubahan yang merepresentasikan langkah-langkah dari satu versi ke versi yang baru.

Pada kasus seperti itu, sebuah framework versioning harus menyediakan mekanisme untuk menentukan kapan dua versi adalah kompatibel.

Tujuan utama dari framework versioning yang menyesuaikan kebutuhan yang telah disebutkan di atas dapat juga ditemukan dalam penelitiannya Klein [Klein et al., 2002]. Tujuan pertama adalah memastikan untuk menyediakan kemudahan akses data pada versi yang berbeda pada sebuah ontologi. Hal ini dapat dicapai, selain dengan ketat diijinkan mengubah hanya pada hal yang mana tidak mempengaruhi interpretasi terhadap data, atau dengan menyediakan translasi di antara versi sehingga kueri dapat ditranslasi kembali dalam rangka mengakses data dalam bentuk yang lama.

Tool untuk Manajemen Perubahan Ontologi: antara lain: OntoView, Protege:Prompt Tab

## 2.2 Peer-to-Peer

### 2.2.1 Definisi dan Karakteristik P2P

Internet dan Web telah membawa kemajuan dalam pertukaran informasi. Tapi pendekatan yang masih digunakan lebih sentralisasi. Situasi ini akan membawa ke kondisi server yang berlebihan beban. Meningkatnya desentralisasi dan otonomisasi organisasi dewasa ini, satu solusi yang menjanjikan bagi interoperabilitas informasi. Pada saat yang bersamaan, pengambilan kembali informasi, pencarian dan pemrosesan informasi berdasarkan pencarian keyword dan lebih sederhana representasi pengetahuan. Hal yang sulit untuk menentukan ketika dua terminologi adalah equivalent tetapi mempunyai arti berbeda. Teknologi semantic web telah memperlihatkan kemampuan untuk membantu memecahkan masalah keberagaman. Model P2P bukanlah suatu teknologi baru, hal itu sudah diperkenalkan pada dua dekade yang lalu. [Milojicic et al., 2002] menyatakan beberapa definisi dari P2P. Dia mengumpulkan beberapa definisi dari para ahli sebagai berikut:

- Kelompok kerja P2P Intel memberikan definisi: berbagi sumber daya komputer dan layanan dengan pertukaran langsung diantara sistem
- Ross Lee Graham menyatakan definisi P2P melalui tiga kunci kebutuhan: (a) P2P mempunyai komputer operasional yang mempunyai kualitas setara server komputer, (b) P2P mempunyai sistem pengalamatan yang independen dari DNS, (c) P2P dapat dan mampu menangani keterhubungan variabel.

P2P adalah bukan jaringan fisik, tetapi P2P lebih merupakan jaringan logik. P2P dapat tersedia di internet, LAN atau jaringan model fisik lainnya. Karakteristik utama dari sistem P2P adalah: Self-organizing, node mempunyai kebebasan untuk mengorganisasi dirinya sendiri ke dalam jaringan. Symmetric communication, node adalah sama atau sederajat, baik layanan permintaan ataupun penawaran. Maka, node perlu mempunyai satu komputer operasional yang mempunyai kualitas setara server komputer, karena setiap node dapat berlaku sebagai satu client dan/atau satu server. Decentralized, P2P tidak mempunyai direktori global atau kontrol terpusat untuk tiap node.

Sebagai tambahan untuk karakteristik utama yang di atas, terdapat karakteristik turunan lainnya, yaitu: otonom, biaya kepemilikan, anonimiti/privasi, kemampuan untuk dikembangkan, ad-hoc connectivity, sistem pengalamatan, dan peer yang dinamis.

Jaringan P2P menawarkan beberapa keuntungan berikut ini:

- P2P tidak memerlukan pengaturan administrasi khusus.
- P2P tidak memerlukan pengaturan administrasi khusus.
- P2P adalah mengorganisasi sendiri (self-organized) dan adaptif. Peer dapat datang dan pergi secara bebas.
- P2P dapat mengumpulkan dan menghubungkan komputasi yang fantastis dan sumber daya storage pada komputer yang melewati internet.

Terdapat hal yang terdistribusi dan desentralisasi. Oleh karena itu, P2P berpotensi untuk mentoleransi kesalahan (fault-tolerant) dan menyeimbangkan beban (load-balance).

Jaringan P2P dapat diklasifikasikan berdasarkan pada beberapa pandangan, dibagi sebagai berikut: (lihat gambar 2.1)

#### **Jaringan P2P berbasis kontrol pada lokasi data:**

- Jaringan yang tidak terstruktur (unstructured network), tidak ada aturan yang hadir yang mendefinisikan di mana data disimpan dan topologi jaringan adalah tidak diinformasikan / acak, contohnya adalah Gnutella [Gnutella, 2003].
- Jaringan berstruktur longgar (loosely structured network), struktur yang berada di atasnya dan lokasi data tidak ditentukan secara tepat/pasti, contohnya Freenet [Freenet, 1999].
- Jaringan berstruktur tinggi (highly structured network), baik arsitektur jaringan dan penempatan data dispesifikasikan dengan tepat, contohnya Chord [Stoica et al., 2001]

### **Jaringan P2P yang berbasis pada direktori dari lokasi objek:**

- Tersentralisasi/terpusat (centralized), satu direktori pusat dari lokasi objek, pengalokasian ID, dan lain-lain adalah dipelihara dalam satu lokasi tunggal, seperti Napster [Napster, 2003]
- Desentralisasi murni (purely decentralized), peer yang secara keseluruhan adalah sama / sederajat, seperti Gnutella
- Sistem campuran (hybrid system), beberapa peer mempunyai tugas sebagai super peer yang melayani permintaan pencarian dari peer yang lain, seperti Freenet.

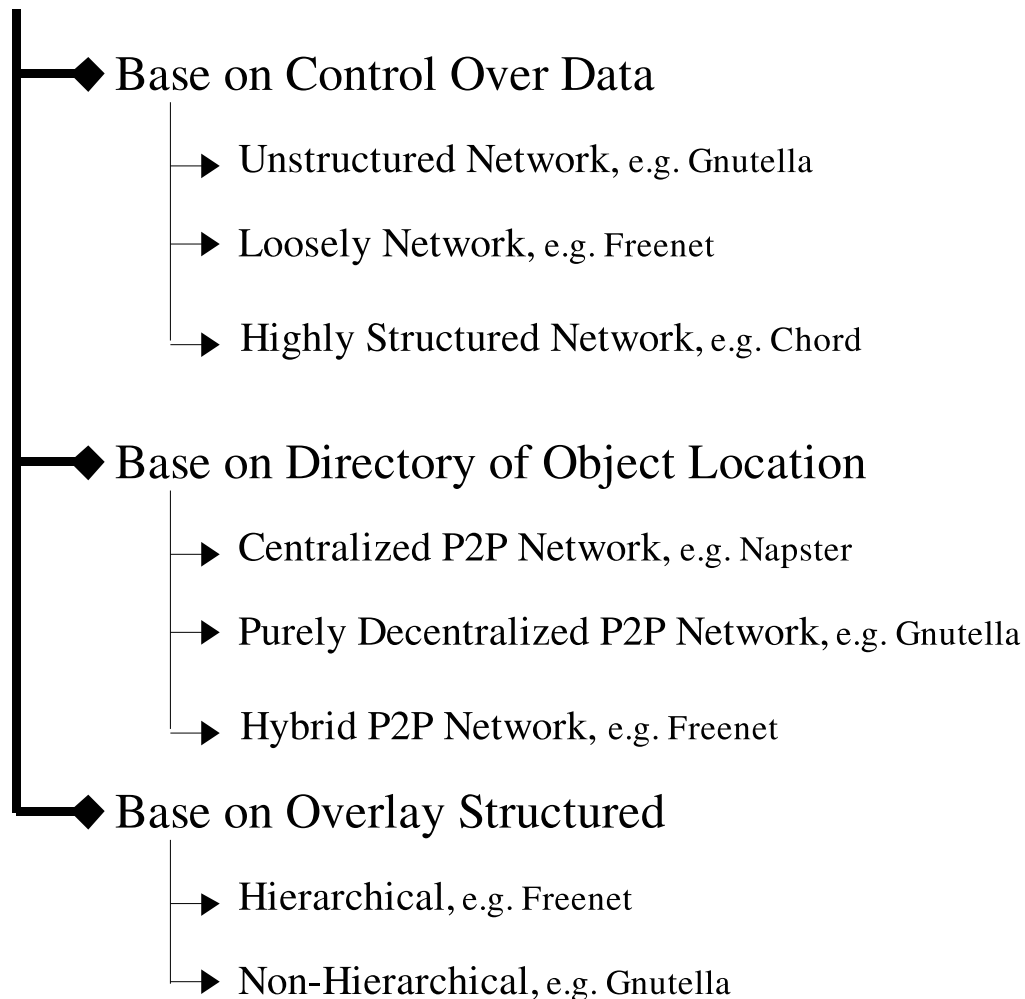
### **Jaringan P2P yang berbasis pada struktur overlay:**

- Sistem hirarki menyediakan kemampuan yang baik akan pengembangannya, kesempatan untuk mengambil keuntungan dari keberagaman node dan efisiensi routing yang tinggi. Seluruh sistem desentralisasi yang campuran dan sedikit murni adalah sistem hirarki.
- Sistem yang non-hirarki menawarkan load balance dan keluwesan yang tinggi (high resilience). Sebagian besar sistem mempunyai overlay yang rata dan sistem yang non-hirarki.

### **Isu utama dari sistem P2P meliputi:**

- Penemuan peer dan data: peer harus dapat mengacu ke dan menempatkan data atau informasi di peer yang lain.
- Pemrosesan kueri: diberikan satu kueri, sistem harus dapat menemukan peer yang memberikan kontribusi data yang relevan dan secara efisien mengeksekusi kueri.
- Interoperabilitas data: ketika sumber data sedang dibagi dalam sistem mengikuti skema yang berbeda atau representasi, peer harus masih dapat untuk mengakses data atau informasi itu, secara ideal menggunakan representasi yang digunakan untuk memodelkan datanya sendiri.
- Ketidakkonsistenan data: jika data direplikasi atau di-cache dalam sistem, satu isu kunci adalah untuk memelihara ketidakkonsistenan di antara yang terduplikasi tersebut.

## P2P Networks



Gambar 2.1: Klasifikasi Jaringan P2P)

### 2.2.2 Peer-to-Peer Discovery

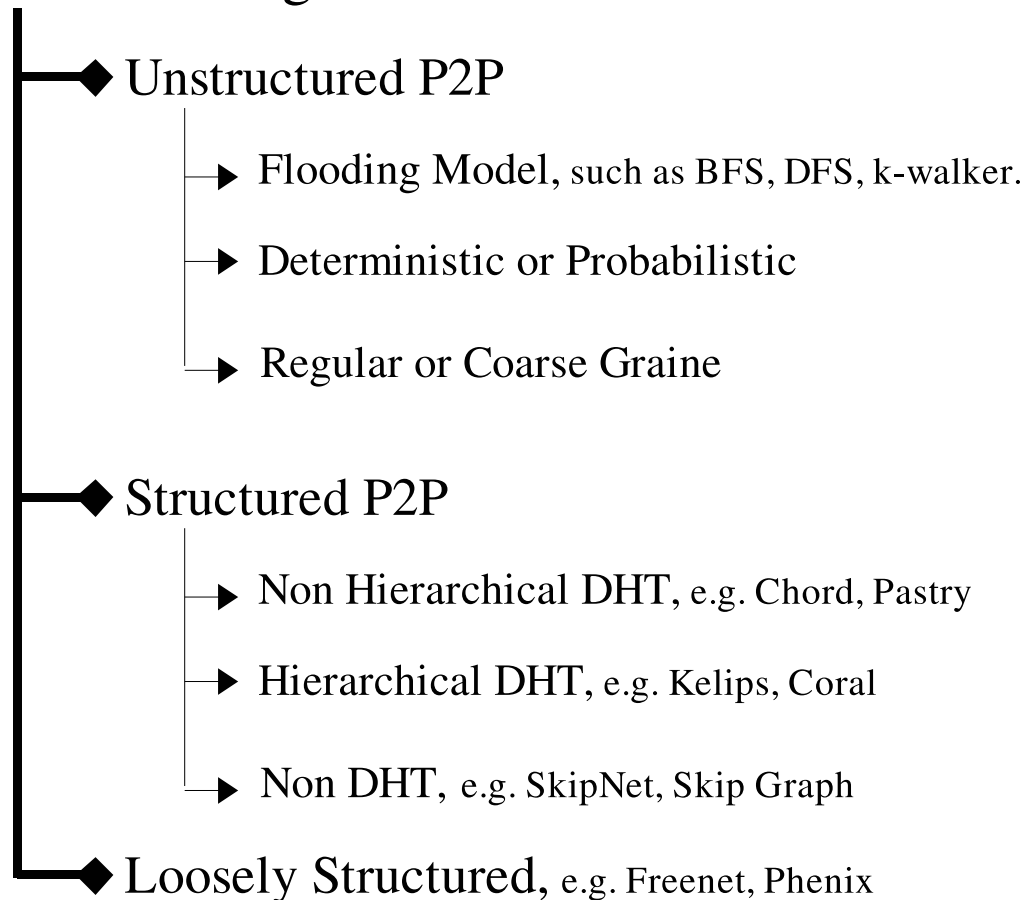
Terdapat banyak isu penelitian dalam komputasi P2P. Sub bab ini memfokuskan pada teknik pencarian dalam jaringan P2P. Pencarian adalah satu usaha untuk menemukan lokasi data yang diinginkan. Sebagian besar sistem P2P yang ada mendukung penemuan objek sederhana dengan kunci atau identifier. Beberapa sistem P2P yang ada dapat menangani kueri keyword yang lebih rumit, yang menemukan dokumen yang berisi keyword dalam kueri. Lebih dari satu salinan dari objek dapat muncul dalam satu sistem P2P. Kemungkinan, terdapat lebih dari satu dokumen yang berisi keyword yang diinginkan. Beberapa sistem P2P tertarik pada satu item data tunggal; lainnya teertarik pada item data keseluruhan atau sebanyak mungkin item data yang mungkin saja dapat memuaskan atau memenuhi kondisi yang diberikan. Sebagian besar teknik

pencarian adalah berbasis forward. Memulai dengan peer yang meminta, satu kueri peer di forward kan (atau di rute kan) ke peer sampai peer tersebut mencapai data yang diinginkan (atau menuju ke data yang diinginkan) tercapai. Untuk mem-forward pesan kueri, tiap peer harus menyimpan informasi mengenai beberapa peer yang lain yang disebut tetangga (neighbour). Informasi dari tetangga ini menyatakan tabel rute dari satu peer.

Fitur yang diinginkan dari algoritma pencarian dalam sistem P2P meliputi hasil kueri yang berkualitas tinggi, status rute minimal dipelihara satu peer, efisiensi rute yang tinggi, load balance, keluwesan untuk kesalahan peer, dan mendukung kueri yang rumit. Kualitas dari hasil kueri adalah application dependent. Pada umumnya, hal itu diukur dengan jumlah hasil dan relevansi. Status rute mengacu ke jumlah tetangga tiap peer yang dipelihara. Efisiensi rute pada umumnya diukur dengan jumlah dari overlay hops per peer. Dalam beberapa sistem, efisiensi rute juga dievaluasi dengan menggunakan jumlah pesan per kueri. Teknik pencarian yang berbeda membuat penggantian yang berbeda diantara karakteristik yang diinginkan tersebut.

Pencarian dalam sistem yang terstruktur tinggi menyediakan jaminan untuk menemukan data yang ada dan bounded data lookup efficiency dalam arti bahwa jumlah overlay hops; bagaimanapun struktur jaringan yang ketat menekan overhead yang tinggi untuk menangani frekuensi peer yang joint-leave (gabung dan meninggalkan). Sistem P2P yang tidak terstruktur adalah sangat lentur bagi peer yang joint-leave, karena tidak ada struktur jaringan yang khusus memerlukan pemeliharaan. Pencarian dalam jaringan yang tidak terstruktur sering berbasis flooding atau variasinya karena tidak ada kontrol atas storage data. Strategi pencarian dalam sistem P2P yang tidak terstruktur adalah pencarian buta atau pencarian terinformasi. Dalam satu pencarian buta seperti *iterative deepening* [Yang and Garcia-Molina, ], tidak ada peer yang mempunyai informasi tentang lokasi dari data yang diinginkan. Pada satu pencarian yang terinformasi seperti peng-indeks-an rute [Crespo and Garcia-Molina, ], tiap peer menyimpan beberapa metadata mengenai lokasi data. Untuk membatasi konsumsi bandwidth secara total, kueri data dalam sistem P2P yang tidak terstruktur dapat diterminasi secara prematur sebelum data yang ada ditemukan, sehingga kueri tidak dapat mengembalikan data yang diinginkan meskipun jika data yang sesungguhnya hadir dalam sistem. Satu jaringan P2P yang tidak terstruktur tidak dapat menawarkan efisiensi rute yang dibatasi oleh karena kekurangan / ketidaksempurnaan struktur. Pencarian dalam sistem struktur longgar bergantung pada struktur overlay dan bagaimana data disimpan. Pada Freenet, pencarian ditujukan melalui petunjuk yang digunakan bagi konstruksi overlay dan penyimpanan data. Pada Symphoni, lokasi data adalah didefinisikan secara

## P2P Searching



Gambar 2.2: Klasifikasi Pencarian di P2P)

tepat tetapi struktur overlay terbentuk secara kebetulan (probabilistically). Pencarian pada Symphony diarahkan melalui penurunan jarak numerik dari sumber kueri ke peer tujuan di mana data yang diinginkan tersimpan/ditempatkan. Sistem struktur yang longgar dapat menawarkan satu keseimbangan transaksi/pertukaran jika sistem tersebut dirancang dengan memadai.

Gambar 2.2 memperlihatkan klasifikasi metode pencarian di lingkungan P2P, sedangkan gambar 2.3 menjelaskan tiap metode dan contoh yang berhubungan.

### 2.2.2.1 Pencarian dalam P2P yang Tidak Terstruktur

Dalam [Jawhar and Wu, 2004], satu sistem P2P yang tidak terstruktur, tidak ada aturan yang secara tegas mendefinisikan di mana data disimpan dan peer yang mana adalah tetangga satu dengan yang lainnya. Untuk menemukan satu item data yang spesifik, penelitian awal seperti Gnutella yang pertama digunakan flooding, yaitu Bread First

Search (BFS) dari graf jaringan overlay dengan batas kedalaman  $D$ .  $D$  mengacu kepada system-wide maximum TTL dari satu pesan dalam ekspresi dari overlay hops. Pada pendekatan ini, peer yang mengkueri mengirim permintaan kueri kepada seluruh tetangganya. Setiap tetangga memproses kueri dan mengembalikan hasil jika data telah ditemukan. Tetangga ini kemudian mem-forward permintaan kueri selanjutnya kepada seluruh tetangganya kecuali peer yang melakukan kueri.

Banyak alternatif skema yang telah diajukan untuk menangani masalah dari flooding yang pertama ditemukan. Penelitian tersebut meliputi iterative deepening [Yang and Garcia-Molina, ], k-walker random walk [Lv et al., 2001], modified random BFS [Kalogeraki et al., ], two-level k-walker random walk [Jawhar and Wu, 2004], intelligent search, directed BFS [Yang and Garcia-Molina, ], local indices based search, routing indices based search [Crespo and Garcia-Molina, ], attenuated bloom filter based search [Rhea and Kubiawicz, ], adaptive probabilistic search [Tsoumakos and Roussopoulos, ], dan dominating set based search [Yang and Wu, 2003]. Semua itu dapat diklasifikasikan sebagai berbasis BFS atau berbasis DFS.

### 2.2.2.2 Pencarian dalam P2P yang Terstruktur

Dalam sistem yang terstruktur secara ketat, hubungan tetangga di antara peer dan lokasi data telah didefinisikan secara ketat. Pencarian dalam sistem seperti itu kemudian ditentukan oleh arsitektur jaringan yang pasti/tertentu. Diantara sistem terstruktur yang ketat, beberapa mengimplementasikan satu tabel hash yang terdistribusi (DHT) dengan menggunakan struktur data yang berbeda. Beberapa sistem P2P DHT mempunyai struktur overlay rata; yang lainnya mempunyai struktur overlay yang berhirarki.

Satu DHT adalah satu tabel hash yang masukan tabelnya terdistribusi diantara peer yang berbeda yang terletak dalam  $s$  lokasi acak. Tiap item data di-hash ke satu kunci numerik yang unik. Tiap peer adalah juga di-hash ke satu ID unik dalam ruang kunci yang sama. Tiap peer bertanggungjawab bagi sejumlah kunci tertentu. Ini berarti bahwa peer yang bertanggungjawab menyimpan kunci dan item data dengan kunci itu atau satu pointer / penunjuk ke item data dengan kunci itu. Kunci dipetakan ke peer yang bertanggungjawab. Algoritma pencarian mendukung dua operasi dasar: lookup(kunci) dan put(kunci). lookup( $k$ ) digunakan untuk menemukan lokasi dari peer yang bertanggungjawab bagi kunci  $k$ . put( $k$ ) digunakan untuk menyimpan satu item data (atau menunjuk ke item data) dengan kunci  $k$  dalam peer bertanggungjawab bagi  $k$ . Dalam satu aplikasi penyimpanan terdistribusi dengan menggunakan DHT,

satu peer harus mempublikasikan berkas-berkas yang asalnya disimpan pada peer itu sebelum peer lainnya mengambil berkas-berkas tersebut. Satu berkas dipublikasikan dengan menggunakan  $put(k)$ .

### **2.2.2.3 Pencarian dalam Non-hierarchical DHT P2P**

Non-hierarchical DHT P2P yang berbeda menggunakan struktur data rata yang berbeda untuk mengimplementasikan DHT. Struktur data rata ini meliputi ring, mesh, hypercube, dan graf khusus lainya seperti graf de Bruijn. Cord [Stoica et al., 2001] menggunakan struktur data ring. Pastry [Rowstron and Druschel, ] menggunakan struktur data berbasis pohon, yang dapat dianggap sebagai generalisasi dari satu hypercube .

### **2.2.2.4 Pencarian dalam Hierarchical DHT P2P**

Seluruh hierarchical DHT P2P mengorganisasi peer ke dalam kelompok atau bagian yang berbeda. Tiap kelompok membentuk overlaynya sendiri. Seluruh kelompok bersama-sama membentuk keseluruhan overlay hirarki. Biasanya, hirarki overlay adalah dua tingkata atau tiga tingkat. Mereka berbeda secara nyata dalam jumlah dari kelompok dalam tiap tingkat, struktur overlay dibentuk oleh tiap kelompok, dan ketika atau tidak peer dibedakan sebagai peer biasa dan super peer / node yang mendominasi. Super peer / node yang mendominasi pada umumnya menyumbang lebih banyak sumber daya komputasi, lebih stabil , dan membutuhkan lebih banyak tugas dalam ruting daripada peer biasa. Diskusi dalam sub bab ini focus pada Kelps [Gupta et al., ] dan Coral [Freedman and Mazières, ].

### **2.2.2.5 Pencarian dalam Non-DHT P2P**

Non-DHT P2P berusaha untuk memecahkan masalah dari DHT P2P dengan menghindari hashing. Hashing tidak menyimpan lokasi / posisi data dan tidak terbuka untuk mengurutkan kueri. Sub bab ini memperkenalkan tiga macam Non-DHT P2P: SkipNet [Harvey et al., ], SkipGraph [Aspnes and Shah, ], dan TerraDir [Silaghi et al., ]. SkipNet dirancang untuk menyimpan data dekat dengan pengguna. SkipGraph bertujuan untuk mendukung pengurutan kueri. TerraDir ditargetkan untuk pencarian name secara hirarki. Pencarian dalam sistem seperti ini mengikuti hubungan bertetangga yang khusus diantara peer.

### 2.2.2.6 Pencarian dalam Loosely Structured P2P

Dalam loosely structured P2P, struktur overlay tidak secara ketat dispesifikasikan. Hal itu dapat dibentuk berbasis pada petunjuk atau dibentuk secara probalistik. Dalam Freenet [Freenet, 1999] dan Phenix [Zhang et al., ], overlay melibatkan ke dalam struktur yang terencana berbasis pada petunjuk atau kesukaan/favorit. Dalam Symphoni [Manku et al., 2003] dan penelitian dalam [Zhang et al., ], overlay dibangun secara probabilistik. Pencarian dalam sistem loosely structured P2P bergantung pada struktur overlay dan bagaimana data disimpan. Dalam Freenet, data disimpan berdasarkan pada petunjuk yang digunakan bagi pembangunan overlay. Sehingga, pencarian dalam Freenet juga berbasis pada petunjuk. Dalam Phenix, overlay dibangun secara independen dari aplikasi. Lokasi data ditentukan oleh aplikasi yang menggunakan Phenix. Sehingga pencarian dalam Phenix adalah bergantung pada aplikasi.

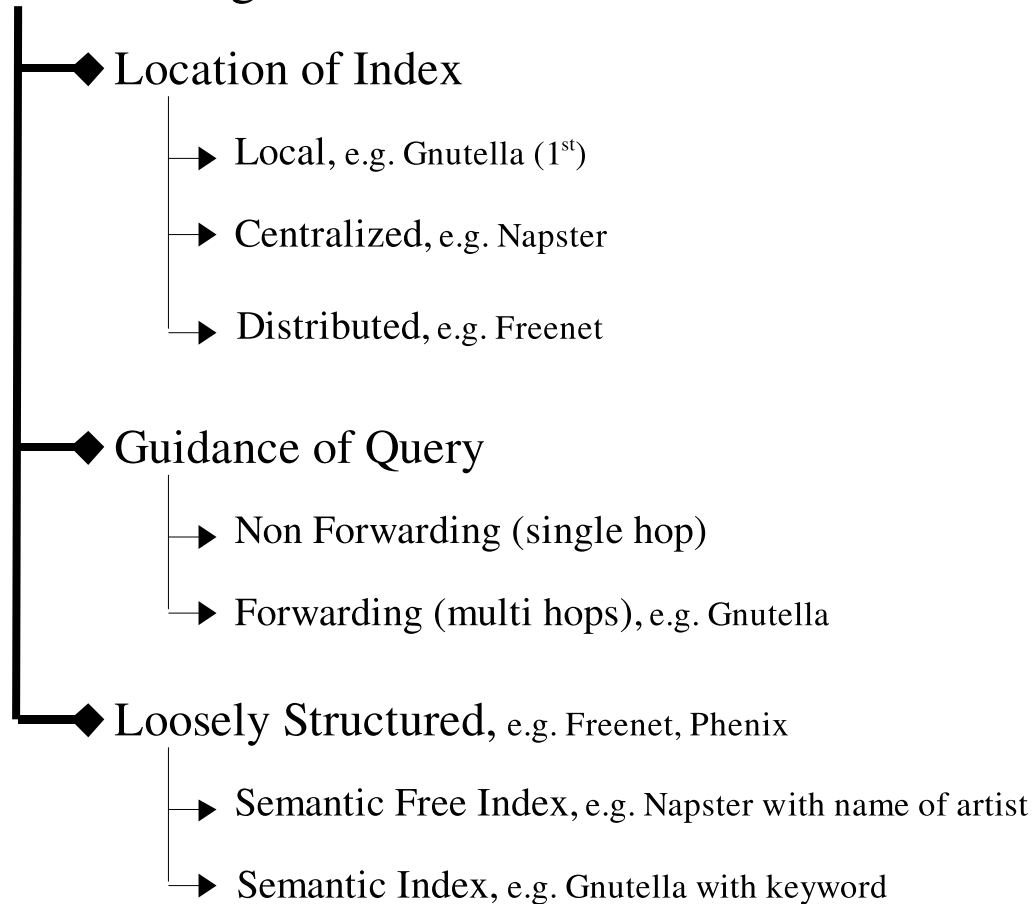
### 2.2.3 Peng-indeks-an dalam P2P

Rekayasa indeks adalah pada jantung dari metode pencarian P2P. Pengindeks-an menggambarkan satu area yang besar dari isu P2P, seperti yang diperlihatkan oleh model Search Index Links (SIL). Seperti Mamber meletakkan SIL, *yang paling penting dari tools untuk pengambilan informasi adalah indeks dari kumpulan terminologi dengan penunjuk ke tempat di mana informasi mengenai dokument dapat ditemukan* [Manber, 1999]. Sen dan Wang menyatakan bahwa satu *jarangan P2P* biasanya berisi koneksi diantara host atau penyinsalan layer aplikasi, daripada untuk transfer data itu sendiri. Hal serupa, kita berkonsentrasi pada indeks dan kueri *tersinyal*. Gambar 2.3 menggambarkan klasifikasi umum dari indeks dalam lingkungan P2P. Paragraf selanjutnya menerangkan model indeks dari P2P.

Indeks P2P dapat menjadi lokal, terpusat atau terdistribusi. Satu indeks lokal, satu peer hanya menyimpan acuan ke datanya sendiri, dan tidak menerima acuan untuk data pada peer nya. Desain Gnutella pada saat awal sekali mengoptimalkan indek lokal. Pada indeks terpusat, satu server tunggal menyimpan referensike data pada banyak peer. Contoh klasik adalah Napster. Dengan indeks terdistribusi, penunjuk menuju target menempati di beberapa peer. Satu contoh yang sangat awal adalah Freenet. Indeks terdistribusi digunakan pada sebagian besar desain P2P dewasa ini, hal ini yang mendominasi survei ini.

Indeks P2P dapat juga diklasifikasikan sebagai non-forwarding dan forwarding. Ketika

## P2P Indexing



Gambar 2.3: Klasifikasi Indexing di Jaringan P2P)

kueri ditunjuk oleh satu indeks no forwarding, kueri tersebut lompat ke peer yang berisi data target dalam satu hop tunggal. Terdapat skema hop-satu semantik dan bebas semantik. Di mana kemudahan untuk dapat dikembangkan ke sejumlah besar peer diperlukan, skema ini telah dilanjutkan/diperluas ke hop-dua. Hal yang lebih umum adalah forwarding P2P di mana jumlah hop bervariasi dengan jumlah total dari peer, bahkan secara logaritmatik. Pertukaran yang berelasi diantara status ruting, lookup latency, update bandwidth peer ke dua bercampur adalah sesuatu yang kritis untuk kemampuan bertahan sistem secara keseluruhan.

Banyak dari pengindeks-an jaringan yang terdistribusi saat ini adalah semantik. Indeks semantik adalah dapat dibaca / dimengerti oleh manusia. Sebagai contoh, hal itu dapat menghubungkan informasi dengan keyword lainnya, satu dokumen, sebuah kunci database atau bahkan sebuah domain administratif. Dan dapat memudahkan untuk menghubungkan objek dengan penyedia jaringan tertentu, perusahaan atau organisasi.

Terdapat beberapa yang semantic free dan indeks semantik. Distributed Hash Table (DHT) telah diusulkan untuk menyediakan referensi yang semantic free dan berorientasi pada data (data-centric). DHT memungkinkan dan memfasilitasi untuk menemukan kunci yang permanen dari objek dalam satu host yang sangat besar, dan berubah ubah kelompoknya.

## Bab III

# PERTIMBANGAN PEMELIHARAAN AGREEMENT

### 3.1 Pendahuluan

Untuk mengatasi keragaman informasi, beberapa pendekatan telah dilakukan, salah satunya adalah dengan menggunakan pendekatan interoperabilitas semantik yang digabungkan dengan P2P [Staab, 2006].

P2P memungkinkan terjadinya pembentukan komunitas yang memiliki kesamaan interest. Dengan terbangunnya group ini maka perbedaan semantik dapat dikurangi. Model ini kerap disebut dengan Semantic Overlay Network (SON). Tetapi pendekatan ini belum memadai sehingga tetap memerlukan jembatan dengan memanfaatkan pendekatan mediasi semantik yang didukung oleh ontologi. Penggunaan ontologi dan P2P telah semakin berkembang dalam beberapa tahun terakhir ini. Karena manajemen pengetahuan dan konten dalam arsitektur P2P lebih mudah dilakukan dibandingkan dengan sistem terbuka penuh.

Pada tingkat arsitektur, jaringan P2P mengizinkan pembuatan komunitas pengguna yang mengorganisasi dirinya sendiri untuk mempublikasikan dan membagi konten. Pada tingkat semantik, tujuan keseluruhan dari semantic web yang terkenal, seperti yang dinyatakan oleh Tim [Berners-Lee et al., 2001], adalah untuk memberikan arti yang didefinisikan dengan baik, membuat komputer dan orang untuk bekerja dalam kooperasi yang lebih baik. Hal itu dapat digunakan untuk penemuan yang lebih efektif, otamasi, keterpaduan, dan penggunaan kembali antar aplikasi. Melanjutkan dengan pola pikir yang sama, Berners-Lee juga mengatakan Saya mempunyai mimpi untuk

Web (dalam hal ini komputer) dapat menjadi mampu menganalisis seluruh data di Web : isinya, link dan transaksi antar manusia dan komputer. Sebuah *Semantic Web*, yang dapat membuat hal ini menjadi nyata, sudah mulai dikenal, tetapi ketika hal itu menjadi nyata dan terjadi, mekanisme hari-ke-hari dari perdagangan, birokrasi, dan kehidupan kita sehari-hari akan ditangani oleh mesin yang berbicara dengan mesin [Berners-Lee et al., 2001].

Bab ini membahas sebuah pendekatan dengan mengacu isu dari mediasi semantik dalam lingkungan yang terdistribusi, khususnya masalah penemuan dan berbagi informasi dalam jaringan P2P yang hirarki, dibangun dari super peer dan peer, akan dibahas, dan juga pembahasan mengenai model usulan pertimbangan untuk pemeliharaan agreement. Akan dimulai dengan mempresentasikan fitur utama dari pendekatan mediasi berbasis agreement. Kemudian akan dikupas aspek dari pengukuran keserupaan/similaritas yang digunakan dalam resolusi perbedaan semantik dari suatu konsep. Selanjutnya, diketengahkan pendekatan semantic agreement, dan hal hal yang langsung berkaitan dengan isu pemeliharaan agreement, mengapa agreement perlu dipelihara dan apa yang menjadi pertimbangan pemeliharaan tersebut. Pada bagian akhir, dibahas beberapa aspek dari pemrosesan kueri berbasis pada pendekatan agreement.

Pembuatan pemetaan semantik di antara sumber informasi yang berbeda adalah titik yang kritis dalam pendekatan integrasi. Banyak pembelajaran yang ada saat ini adalah berbasis pada ide bahwa pemetaan (*mapping*) dapat dengan mudah dibuat oleh perancang yang ahli ketika skema dari sumber informasi yang berbeda dikombinasikan dan diintegrasikan. Seperti kebalikannya, identifikasi dari keterhubungan semantik di antara sumber informasi yang berbeda adalah satu masalah yang sulit walaupun pada sistem terpusat.

Selain itu, identifikasi dari keterhubungan semantik di antara sumber informasi yang terdistribusi harus memikirkan beberapa karakteristik, meliputi (1) keragaman model data yang digunakan dalam merepresentasikan sumber, (2) otonomi dari sumber dalam tipe, kualitas dan kuantitas dari informasi meta yang dibuat tersedia, dan (3) kegiatan / perilaku dinamis dari sumber. Pengintegrasian dari sumber informasi telah menjadi satu area penelitian yang penting untuk beberapa tahun ini. Metode yang ada saat ini dapat secara umum dikategorikan menjadi: Pendekatan manual memberikan petunjuk metodologi untuk mengidentifikasi dan mengekspresikan mapping. Pendekatan semi otomatis mengusulkan mapping dan membiarkan pengguna mendefinisikan mapping akhir. Metode otomatis berusaha untuk menemukan mapping tanpa intervensi

dari pengguna.

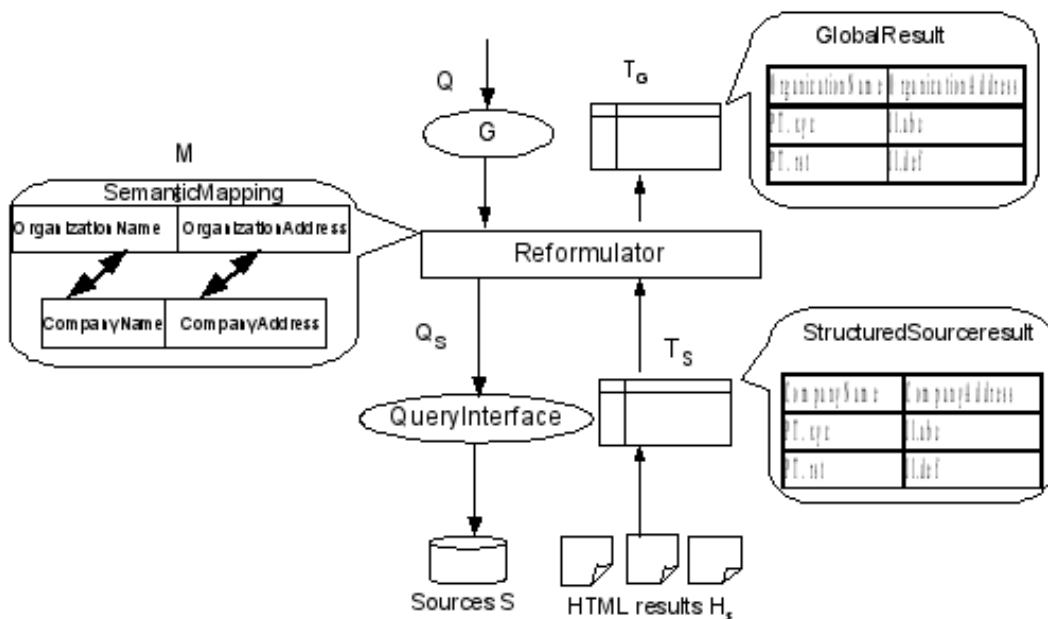
Untuk membuat mapping di antara konsep yang berelasi secara semantik dari sumber-sumber yang beragam, sejumlah kriteria yang berbeda dapat digunakan, diantaranya yang paling jelas adalah mencocokkan nama atau label dari elemen skema. Metode perbandingan linguistik dapat digunakan untuk mencocokkan nama atau label dari elemen skema dengan bersandar pada semantik latentnya. Pada tingkat yang lebih tinggi lagi, struktur dari informasi dapat digunakan sebagai satu kriteria (misalnya atribut dari satu kelas) untuk mengidentifikasi konsep yang berelasi. Bagaimanapun juga, pendekatan integrasi yang hanya bersandar pada struktur informasi memfasilitasi untuk menyelesaikan seluruh ambiguitas diantara terminologi skema dan elemen. Beberapa teknik integrasi yang ada dewasa ini menekankan penggunaan informasi tambahan untuk memecahkan inkonsistensi dalam arti dari sumber-sumber yang berbeda. Informasi tambahan pada dasarnya berisi relasi semantik antara term. Relasi semantik ini dapat menjadi (1) pengetahuan pendahulu atau asumsi rancangan yang berasosiasi dengan term atau (2) didapatkan dari analisis yang melelahkan dari instan yang berkoresponden dengan term dari sumber informasi.

Seperti yang diungkapkan di atas, salah satu karakteristik P2P adalah dinamis dan otonom, dan masing masing peer dapat mendefinisikan skema lokal yang berbeda-beda, sehingga isu keberagaman muncul di sini. Gambar III.1 berikut adalah ilustrasi contoh yang diambil dari [Wicaksana and Guritno, 2005] yang dapat memberikan gambaran sebuah sistem P2P yang dinamis, otonom dan heterogen.

Sebuah perusahaan A bergerak dalam bidang perakitan komputer yang disimbulkan sebagai peer P61 ingin mencari pegawai baru untuk staf teknisi dan penguji kualitas. Staf penguji kualitas akan dikirim ke pelatihan untuk dapat menggunakan perangkat pengujian tertentu. Group 1 yang terdiri dari SP5 dan SP6 adalah grup dari pemberi kerja. Group 2 yang terdiri dari SP1, SP2 dan SP3 adalah group pencari kerja. SP3, SP4 dan SP5 tergabung ke dalam Group 3, adalah sebuah grup pelaksana pelatihan. Sebuah super peer dapat tergabung kedalam lebih dari satu group, sebagai contoh SP5 adalah anggota dari Group 3 dan Group 1.

Asumsikan sebuah perusahaan A (P61) bergabung dengan SP6, dan mengirimkan skema dan metadatanya. Secara mendasar, mekanisme pencarian dalam arsitektur ini akan dilakukan dengan mencari peer mana yang paling sesuai untuk menjawab query. Kapanpun perusahaan A ini melakukan query, dia akan mengirim ke SP6. Pertamakali, SP6 akan melihat kepada Index-Member dan Index-Cache di SP6. Misalkan, jawaban

dimungkinkan diberikan oleh peer P21 untuk informasi pencari kerja, dan P31 untuk informasi pelatihan. Sayangnya, kedua peer tersebut (P21 , P31) bukanlah anggota SP6, dan tidak ada 'cache' di SP6 untuk menhandel query tersebut. Oleh karena itu, Index-Related-SuperPeer akan digunakan untuk mencari, asumsikan ada informasi tentang SP2 dan SP3 pada SP6. Sehingga query akan di transfer dari SP6 ke SP3 ke SP4 ke SP2 baru mencapai P21 untuk mendapatkan informasi tentang pencari kerja. Sementara untuk informasi taining akan melalui rute SP6-SP3-P31. Hasil pemberian jawaban terhadap query tersebut dan peer mana yang dapat menjawab akan disimpan pada cache SP6. Jawaban ini bisa datang lebih dari satu peer, di mana permasalahan untuk union dan join respon peer adalah kasus tersendiri. Dengan mekanisme ini maka pendekatan dengan 'broadcast' pada 'pure' P2P akan dapat dihindari sehingga akan mengurangi penggunaan bandwidth dan meningkatkan kecepatan dan ketepatan pencarian respon terhadap query.



Gambar 3.1: Contoh P2P Marvick

Di dalam pendekatan *semantic agreement* [Wicaksana, 2006], terdapat dua buah lingkungan, yaitu lingkungan berbagi informasi (*information sharing*) yang mendefinisikan *common ontology*, yang merupakan sumber daya untuk berbagi data dan mengendalikan interaksi peer dan lingkungan yang ke dua adalah lingkungan *peer peer environment* yang digunakan untuk menggambarkan dan mempublikasikan konten lokal dari sumber data. Sebuah *peer* (PP) menyediakan informasi kepada komunitas, memungkinkan akses

ke bagian dari datanya melalui sebuah skema ekspor (skema lokal / ontologi lokal) dan mendefinisikan *agreement* untuk merelasikan lokal datanya ke *common ontology*.

Sebuah *peer* dapat mengirimkan (submit) kueri ke dirinya sendiri dan ke *peer* yang lain. Sebuah *peer* mengandung beberapa elemen: (1) sebuah skema ekspor yang merepresentasikan *view* pada data berbagi (*shared data*) yang dapat diakses oleh peer-peer yang lain, (2) basis data lokal dan (3) informasi *agreement* untuk membuat link antara skema ekspor dan *common ontology*.

Pada contoh ilustrasi P2P yang terdapat pada gambar III.1, jelas terlihat masing masing *peer* dapat mendefinisikan skema lokal yang berbeda beda. Skema lokal atau data lokal merepresentasikan data berbagi (*shared data*) pada *peer*. Data lokal dapat disimpan dalam model data tradisional (relasional atau obyek) atau format XML. Untuk dapat berukar data antara *peer* yang satu dengan *peer* yang lain diperlukan suatu pivot atau 'pengatur' pertukarannya, karena seperti yang sudah sering disebutkan di muka bahwa masing masing *peer* dapat mempunyai skema dan konten informasi yang berbeda. Dan fungsi pivot diambil oleh *common ontology*. Sebuah ontologi digunakan untuk meng- *capture* semantik dari konsep yang digunakan oleh aplikasi yang berbeda. Ontologi adalah kosakata yang dibangun dari term dan relasi diantaranya. Di sinilah peran *agreement* muncul, yaitu ketika konflik semantik timbul yaitu ketika dua sistem tidak menggunakan interpretasi yang sama terhadap suatu informasi. Untuk mengatasi perbedaan semantik di antara konten dari *peer*, maka *agreement* dibuat untuk merepresentasikan hasil dari pencocokan atau pemetaan konsep dari satu *peer* ke konsep dari *peer* yang lain.

Dalam lingkungan yang dinamis dan otonom, seperti pada P2P, administrator harus secara terus menerus memonitor *common ontology*. Bila dilakukan monitoring secara manual sangat mahal dan tidak dapat terskala dengan baik jika jumlah *peer* semakin meningkat. Oleh karena itu sangat penting untuk mengembangkan suatu cara untuk secara otomatis melakukan monitoring dan memperbaiki *agreement* yang ada. Dikatakan dinamis karena informasi dan juga skema lokal yang terdapat pada *peer* dapat berubah sesuai dengan kebutuhan dan perkembangan yang ada. Sejalan dengan itu diperlukan pemeliharaan terhadap *agreement* yang sudah terbentuk sebelumnya.

Pada bab ini, juga dijelaskan mengenai isu mediasi semantik untuk mendukung berbagi informasi dalam jaringan P2P. Beberapa karakteristik dari lingkungan terdistribusi yang dimasukkan dalam pendekatan *semantic agreement* adalah (1) kenyataan bahwa konstruksi dari skema global untuk mengkombinasikan seluruh informasi berbagi ada-

lah tidak realistis dan sangat tidak mungkin untuk dilakukan, (2) perilaku dinamis dari peer, seperti *peer* dapat secara bebas dan teratur gabung (*join*) atau meninggalkan (*leave*) jaringan, dan (3) kenyataan bahwa peer dapat menggunakan konsep yang berbeda dengan variasinya pada semantik.

Pendekatan mediasi yang terdapat dalam Wicaksana mengikuti arsitektur mediator/wrapper. Wrapper pada dasarnya direpresentasikan oleh sumber data lokal dan skema ekspor yang biasa membungkus data berbagi. Peran mediator dimainkan sebagian oleh common ontology yang disediakan dan dikontrol oleh super peer dan sebagian semantic agreement dibuat oleh peer untuk mengizinkan peer lain memvisualisasi dan meminta konten data nya. Semantic agreement digunakan untuk menangani interoperabilitas, seperti menemukan sumber dan kueri. Kebutuhan dasar dari agreement adalah bagaimana merepresentasikan deskripsi konteks dari sumber dan melakukan agreement.

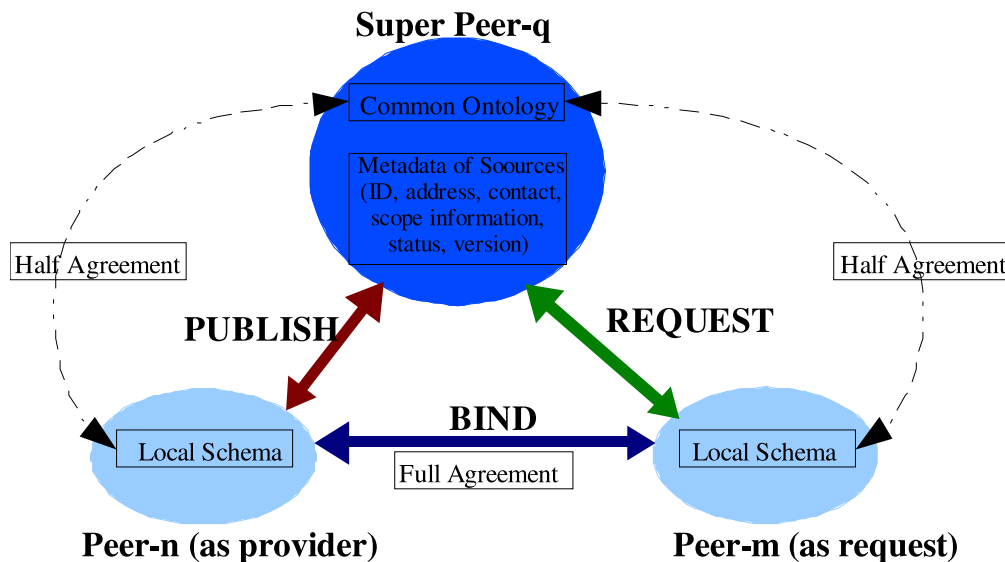
## 3.2 Fitur Pendekatan Semantic Agreement

Bagian ini menyajikan (1) gambaran dari menemukan dan mencocokkan sumber informasi, (2) definisi notasi dan model representasi yang digunakan dalam bab ini, khususnya mendetilkan model XML dan RDF/OWL yang digunakan dalam deskripsi konten, dan (3) komponen utama dari pendekatan mediasi berbasis agreement

### 3.2.1 Penemuan dan Pencocokan dalam Lingkungan P2P

Pada bab 2 sudah membahas arsitektur P2P yang berbeda. Mengacu bahasan sebelumnya, pada bab ini mengulang kembali arsitektur P2P hybrid yang dibangun dari super peer yaitu server yang mempunyai kemampuan tinggi. Super peer dihubungkan ke dan mengelola simple peer dengan kemampuan dan sumber daya yang lebih sedikit. Dalam pendekatan pencocokan semantik yang dikembangkan oleh [Wicaksana, 2006], super peer digunakan untuk mengatur sumber daya yang biasa digunakan untuk berbagi informasi. Peer merepresentasikan sumber data dan mensubmit kueri untuk mempublikasikan dan berbagi data. Gambar 3.2 mengilustrasikan langkah yang berbeda yang dilakukan oleh peer ketika peer men-submit atau menerima kueri. Operasi dari pendekatan agreement dilakukan dalam dua fase. Pada fase pertama, common ontology dan context metadata didefinisikan dan disimpan dalam satu atau lebih super peer yang beraksi sebagai koordinator bagi pertukaran data dan kegiatan berbagi. Fase kedua adalah didedikasikan untuk pemrosesan kueri dan pesan berbagi data yang dieksekusi oleh peer. Terdapat dua tipe pesan: Publish dan Request. Super peer menerima

dan memproses pesan Publish dan Request dari simple peer. Seluruh peer berinteraksi dalam satu proses yang umum dari berbagi dan pertukaran informasi seperti berikut ini:



Gambar 3.2: Penemuan dan Pencocokan di Lingkungan P2P [Wicaksana, 2006])

1. **Publish:** simple peer, disebut provider peer (*PP*), menyediakan kontennya dan dapat mengakses data lain dari peer. Ketika satu provider peer bergabung dengan jaringan, PP mengirim satu pesan Publish ke super peer dan melakukan inisiasi proses pencocokan untuk memulai hubungan diantara konsep lokal dan ontologi. Ketika super peer menerima pesan Publish dari simple peer, super peer menyimpan identitasnya dan pemetaan yang telah dibuat.
2. **Request:** satu peer mengirim satu pesan Request ke satu atau lebih super peer untuk menemukan atau melokasikan sumber data yang relevan dengan kuerinya. Ketika satu super peer menerima satu pesan Request, super peer mencari metadata *PP* yang tersedia dan mengembalikan ke pengirim satu daftar kandidat sumber data. Pada saat menerima daftar kandidat sumber dari super peer, provider menginisiasi satu proses untuk menghitung similaritas diantara skemanya dan skema dari kandidat sumber.
3. **Bind:** langkah ini digunakan untuk membuat korespondensi diantara konsep yang terkandung pada skema dari dua peer. Ketika satu peer yang meminta menerima daftar kandidat sumber dari super peer, peer tersebut berusaha untuk menggabungkan (Bind) konsepnya dengan satu atau lebih kandidat peer itu. Pengga-

bugan berbasis pada pemetaan ontologi dimulai dengan pesan Publish dalam langkah Publish di atas.

### 3.2.2 Model untuk Merepresentasikan Informasi Peer

Dalam [Wicaksana, 2006] digunakan dua model representasi, yaitu XML dan RDF/OWL untuk menggambarkan konten dari peer. XML digunakan dalam deskripsi dari sumber data lokal. Walaupun pendekatan yang diusulkan dapat mengakomodasi model data seperti model relasional dan objek, contoh yang digunakan untuk mengilustrasikan ide yang didiskusikan semuanya berbasis pada deskripsi XML dari informasi lokal. Di bawah ini akan diberikan satu deskripsi singkat dari XML dan RDF/OWL.

Extensible Markup Language (XML) adalah satu format teks sederhana, sangat fleksibel dan juga memperlihatkan satu peran yang penting yang terus menerus meningkat dalam pertukaran data yang sangat bervariasi di web dan di tempat yang lain. Tiap dokumen XML mempunyai baik struktur logikal dan fisikal. Secara fisik, dokumen dibuat dari unit yang disebut entitas. Satu entitas dapat mengacu ke entitas lain menyebabkan disisipkannya ke dalam dokumen. Satu dokumen dimulai dalam satu *root* atau entitas dokumen. Secara logis, dokumen dibangun dari deklarasi, elemen, komentar, referensi karakter, dan instruksi pemrosesan, seluruhnya di-indikasikan dalam dokumen oleh explicit markup. XML namespaces menyediakan satu metode yang sederhana untuk mengkualifikasi elemen dan nama atribut yang digunakan dalam dokumen XML melalui pengasosiannya dengan namespaces yang di-identifikasi oleh referensi URI. Sebuah nama yang dipanjangkan adalah sepasang yang terdiri dari nama namespace dan nama lokal.

Resource Description Framework (RDF) adalah satu bahasa berbasis web yang secara khusus diperuntukan untuk mengekspresikan metadata sumber daya yaitu deskripsi dari sumberdaya web dengan properti seperti judul, pengarang, modifikasi data, informasi akses, status ketersediaan dan lain sebagainya. Secara lebih umum, RDF dapat juga digunakan untuk merepresentasikan informasi pada hal hal lain yang dapat diidentifikasi pada web. Di antara karakteristik utama dari RDF, fitur yang paling penting bagi pendekatan yang dikembangkan dalam [Wicaksana, 2006] adalah bahwa RDF menyediakan satu kerangka kerja untuk mengekspresikan metadata sumberdaya dan informasi semantik yang terkait untuk mengijinkan pertukaran diantara aplikasi. RDF menggunakan identifier web yang unik (URI) untuk mengidentifikasi sumberdaya dari yang terkandung dalam domain yang ada. Sintaks dasar terdiri dari satu elemen

rdf:Description, yang berisi satu himpunan elemen properti. Mengenai rdf:attribute mengidentifikasi, sumberdaya mana yang digambarkan. Sementara properti rdf:type digunakan untuk mengekspresikan bahwa sebuah sumberdaya adalah anggota dari kelas yang diberikan atau seperti properti tipe instan dari link yang digunakan oleh jaringan semantik dan sistem frame. Terdapat banyak variasi dari singkatan dalam sintaks RDF. Untuk menghindari konflik penamaan diantara kosakata yang berbeda, RDF menggunakan satu XML namespaces untuk tiap kosakata. Skema RDF mengizinkan pengguna untuk membuat skema dari kelas dan properti standar. Untuk tujuan tersebut, spesifikasi mendefinisikan sejumlah kelas dan properti yang mempunyai semantik pasti. Satu kelemahan yang signifikan dari RDF adalah bahwa RDF tidak menspesifikasikan satu fitur inklusi skema.

Ontology Web Language (OWL) dirancang untuk digunakan oleh aplikasi yang membutuhkan proses konten dari informasi lebih daripada hanya menyajikan informasi bagi manusia. OWL memfasilitasi kemampuan interpretasi mesin yang lebih besar dari konten web kemudian OWL didukung oleh XML, RDF dan Skema RDF (RDF-S) dengan menyediakan kosakata tambahan dengan semantik formal. OWL menggunakan baik URI untuk penamaan dan kerangka kerja deskripsi bagi web yang disediakan oleh RDF untuk menambah kemampuan kepada ontologi, seperti openness, extensibility, scalability dan terdistribusi pada banyak sistem. OWL dibangun pada RDF dan skema RDF dan menambah lebih banyak kosakata untuk menggambarkan properti dan kelas diantara lainnya, relasi diantara kelas (misal, disjointness), kardinalitas (misal, exactly one), equality, properti tipe yang lebih kaya, karakteristik dari properti (misal, simetri), dan daftar spesifikasi kelas. OWL mempunyai tiga subbahasa yang ekspresif: OWL Lite, OWL DL, dan OWL full.

### 3.3 Isu Arsitektur Pendekatan Semantic Agreement

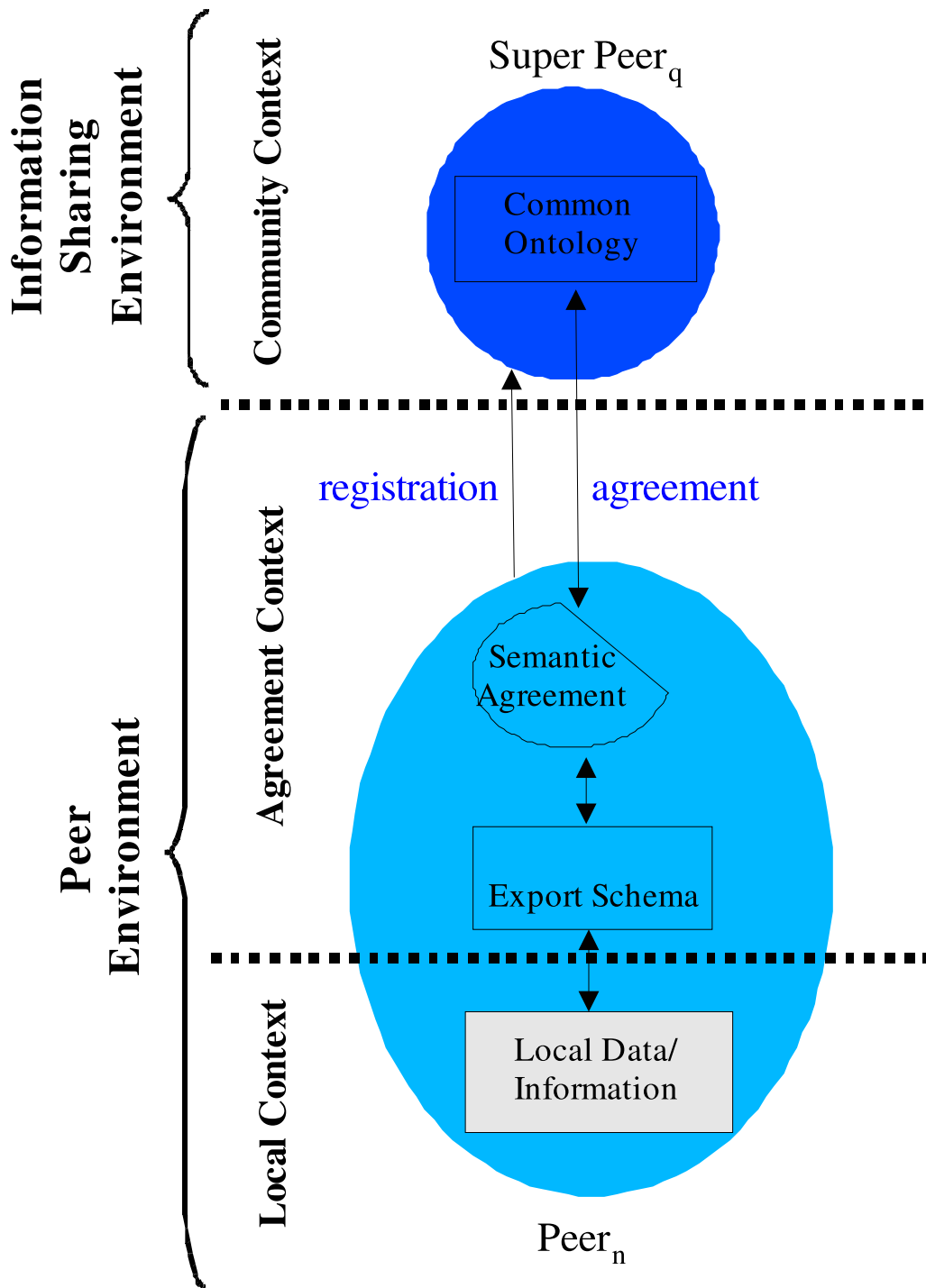
Gambar 3.3 menggambarkan keseluruhan arsitektur dari pendekatan agreement. Pendekatan tersebut terdiri dari lingkungan berbagi informasi yang berisi sumber daya untuk berbagi data dan mengendalikan interaksi peer, dan lingkungan peer yang digunakan untuk menggambarkan dan mempublikasikan konten lokal dari sumber data. Terdapat dua tipe peer: (1) super peer berasosiasi dengan lingkungan berbagi informasi, dan (2) simple peer adalah digambarkan oleh lingkungan peer.

Fitur arsitektur lainnya yang tergambar pada gambar 3.3 adalah konteks yang digunak-

an untuk menggambarkan himpunan data dan yang terutama untuk menyimpan asumsi yang digunakan dalam komponen yang berbeda dari pendekatan. Deskripsi konteks dapat dalam berbagai bentuk seperti ringkasan bahasa alami, spesifikasi yang semi formal. Pada pendekatan [Wicaksana, 2006], seluruh konteks dan deskripsi diekspresikan dalam RDF/OWL.

### 3.3.1 Lingkungan Berbagi Informasi

Lingkungan berbagi informasi direpresentasikan oleh Community context untuk memodelkan aktifitas dari super peer. Lingkungan tersebut berisi dua tipe data: sebuah common ontology dan metadata atau bookkeeping record yang digunakan oleh kedua tipe peer. Common ontology, seperti yang didefinisikan oleh Gruber [Gruber, 1995], sebuah ontologi adalah satu spesifikasi eksplisit dari sebuah konseptualisasi entitas dunia nyata dari satu domain aplikasi. Sebuah ontologi digunakan untuk meng-capture semantik dari konsep yang digunakan oleh aplikasi yang berbeda. Ontologi adalah kosakata yang dibangun dari term dan relasi diantaranya. Beberapa bahasa dan format dapat digunakan dalam menggambarkan ontologi, termasuk bahasa alami dan metode formal seperti RDF, OWL dan bahasa deskripsi lainnya. Terdapat dua tipe spesifikasi dari ontologi. Spesifikasi yang lebih luas bertujuan untuk menggambarkan seluruh himpunan yang mungkin dari konsep yang digunakan dalam domain aplikasi. Sementara spesifikasi yang intensif bertujuan untuk mendefinisikan sebuah himpunan minimum dari konsep dan relasi yang akan diperluas oleh konsep dan relasi baru melalui spesialisasi dari komponen pembuka (awal). Yang terdahulu tidak cukup untuk membuka dan lingkungan yang dinamis karena jumlah konsep yang dibutuhkan untuk menggambarkan dapat berkembang secara cepat ketika jumlah peer dalam jaringan meningkat. Spesifikasi intensif dari ontologi diadopsi dalam pendekatan semantic agreement yang dikembangkan dalam [Wicaksana, 2006]. Dalam pendekatan tersebut, ontologi dalam community context dibangun menggunakan pendekatan local as view (LAV). Ontologi dapat di-view sebagai sebuah skema virtual yang dibuat dari konsep generik yang dapat digunakan oleh sumber lokal untuk mendefinisikan pemetaan semantik view express pada konsep ontologi. Setiap konsep dari common ontology dapat dihubungkan ke satu atau lebih konsep lokal melalui pemetaan semantik. Dalam hal itu, spesifikasi dari konsep dari ontologi seperti definisi yang terencana dari kelas dalam basis data tradisional yang mendefinisikan dua properti atau aturan yang digunakan untuk mengasosiasikan objek dengan definisi. Kemudian, konsep ontologi generik tidak mempunyai populasi atau instant. Ontologi generik mempunyai populasi virtual yang dihitung atau diba-



Gambar 3.3: Representasi Model *Agreement*)

ngun dari populasi dari konsep lokal yang berkorespondensi dihubungkan ke ontologi generik via proses semantic agreement. Diberikan sebuah konsep common ontology generik CO populasinya didefinisikan oleh:

$$population(CO) \equiv \cup population(PP_i : CO)$$

dimana  $PP_i : CO$ ,  $population(PP_i : CO)$  adalah konsep dari  $PP_i$  dan populasinya yang bersesuaian.

### 3.3.2 Pengukuran Similaritas

Untuk mendefinisikan semantic agreement diantara onologi dan konten dari peer, kita harus menghitung ukuran similaritas semantik diantara konsep yang digunakan oleh pemakai yang berbeda beda. Hal ini dapat dilaksanakan dengan pertama kali melakukan pencocokan label dari konsep yang berbeda untuk mengidentifikasi relasinya atau keterhubungannya. Setiap konsep direpresentasikan oleh satu hirarki dari label secara terminologi yang berisi informasi struktur ataupun semantik.

Similaritas semantik adalah suatu kasus khusus dari keterhubungan semantik. Keterhubungan semantik adalah pertanyaan dari bagaimana hubungan dari dua konsep. Dalam [Warin, 2004], Warin menunjukkan perbedaan antara similaritas semantik dan keterhubungan semantik. Similaritas semantik hanya merefleksikan relasi IS-A (hypernym/hyponim). Sebagai contoh, walaupun mobil dan gasolin tidak dihubungkan dengan relasi IS-A, tetapi dua kata itu sangat dekat hubungannya (keterhubungan), karena gasolin adalah bahan bakar yang digunakan oleh sebagian besar mobil. Mobil dan sepeda serupa secara semantik, bukan karena mereka mempunyai roda dan stir, tetapi karena mereka keduanya adalah IS-A dari kendaraan. Maka, relasi antara similaritas semantik dan keterhubungan secara semantik adalah asimetrik.

Dalam pendekatan semantic agreement [Wicaksana, 2006] yang dikembangkan dalam, similaritas semantik dihitung dengan dua cara. Satu cara untuk merefleksikan hanya label ketika dua konsep dicocokkan, secara independen dari posisi mereka dalam hirarki konsep. Metode ini yang menggunakan WordNet untuk menghitung nilai yang cocok. Cara lain adalah untuk menghitung pengukuran semantik dengan memasukan tidak hanya semantik turunannya atau yang implisit dari label, tetapi juga struktur internalnya dan posisinya dalam taksonomi konsep (struktur eksternal).

### 3.3.2.1 Pencocokan Label

Dua langkah utama [Giunchiglia and Zaihraye, ] dilakukan dalam pencocokan label. Pertama, langkah pemrosesan bahasa adalah digunakan untuk mentrasform label ke dalam kata kata yang terpisah. Sebagai contoh, singkatan dibuat kepanjangannya dan akronim diganti dengan teks yang lengkap dalam langkah ini. Ke dua, langkah analisis linguistik adalah digunakan untuk mencocokkan label dengan menentukan relasi antara mereka dengan menggunakan thesaurus seperti WordNet. Beberapa algoritma dan metode yang diusulkan dalam literatur telah diselesaikan pada tahap kelas dan properti. Nilai threshold digunakan untuk memutuskan jika dua label serupa atau tidak. Pada akhirnya, relasi semantik (seperti relasi sub atau super) antara konsep yang berasosiasi dengan label dimasukkan.

**Menghitung Nilai Similaritas dari Dua Label** Beberapa metode Budanitsky [Budanitsky and Hirst, ] dan Warin [Warin, 2004] dapat digunakan untuk menghitung Simlabel. Beberapa metode berbasiskan WordNet, yaitu sebuah basis data jaringan semantik bahasa Inggris yang dibuat oleh Universitas Princeton. Bangunan dasar dalam WordNet adalah synset. Synset adalah himpunan sinonim yang menandai konsep yang sama. Setiap konsep mempunyai satu atau lebih deskripsi. Synset diinterkoneksi oleh link relasional yang berbeda, seperti hypernim (Is-a-of), meronim (Is-a-Part-of), antonim (is-an-opposite-of) dan yang lainnya.

Dalam penelitian yang sudah dilakukan dalam [Wicaksana, 2006] digunakan persamaan pencocokan yang diusulkan oleh Jiang-Conrath (jnc) untuk menghitung nilai similaritas [Warin, 2004]. Persamaan ini menggunakan ide dari information content (IC) dan jnc berbasis pada WordNet. Persamaan similaritas jnc didefinisikan sebagai berikut:

$$Sim_{label} = sim_{jcn}(c_1, c_2) = max\left[\frac{1}{dist_{jcn}(c_1, c_2)}\right] \quad (III.1)$$

dimana,  $dist_{jcn}(c_1, c_2) = IC_{c_1} + IC_{c_2} - 2 * IC(LCS_{(c_1, c_2)})$ .  $LCS$  adalah node yang paling bawah yang termasuk atau mendominasi  $c_1, c_2$ . Sebagai contoh, *animal* adalah node umum paling bawah dari *cat* dan *dog*.

Nilai information content dalam III.1 dikalkulasi oleh :

$$IC_c = -log p(c) \quad (III.2)$$

dimana  $c$  adalah sebuah konsep dalam WordNet dan  $p(c)$  adalah probabilitas dari menjalankan/menemukan  $c$  dalam corpus yang diberikan.  $p(c)$  didefinisikan oleh :

$$p(c) = \frac{\sum_{w \in W(c)} \text{count}(w)}{N} \quad (\text{III.3})$$

dimana  $W(c)$  adalah himpunan dari kata (kata benda) dalam corpus yang sense-nya adalah dihitung oleh konsep  $c$ , dan  $N$  adalah jumlah keseluruhan dari token kata (kata benda) dalam corpus yang juga terdapat dalam WordNet.

**Representasi Hasil dari Pencocokan Label** Pencocokan label dilakukan berpasangan antara label ontological (super peer) dan label lokal (peer). Hasil dari pencocokan label direpresentasikan dengan tupel berikut ini:

$$\langle N_{SP}^i, Rel, N_{PP}^j \rangle \quad (\text{III.4})$$

dimana  $i = 1..i_{max}$ ,  $j = 1..j_{max}$ ;  $N_{SP}^i$ ,  $i_{max}$  berturut turut adalah label dari konsep ke  $i$  dan jumlah label pada super peer  $SP$  ;  $N_{PP}^j$ ,  $j_{max}$  adalah label dari konsep ke  $j$  dan jumlah dari label pada peer  $PP$ ;  $Rel$  adalah relasi konsep dan dapat berupa  $\subset, =, \supset$ .

Langkah pertama adalah menghitung similaritas label antar pasangan dengan menggunakan persamaan III.4, hasil dari langkah ini di representasikan dengan baris berikut ini:

$$\langle N_{SP}^i, N_{PP}^j, Sim_{label} \rangle \quad (\text{III.5})$$

where  $i = 1..i_{max}$ ,  $j = 1..j_{max}$ ;  $N_{SP}^i$ ,  $i_{max}$  berturut-turut adalah label dari konsep ke  $i$  dan jumlah label pada super peer  $SP$  respectively;  $N_{PP}^j$ ,  $j_{max}$  adalah label dari konsep ke  $j$  dan jumlah label pada peer  $PP$ ;  $Sim_{label} \in [0, 1]$  adalah sebuah bilangan nyata yang menspesifikasikan nilai dari label.

Langkah ke dua adalah menentukan himpunan kandidat peer dari label yang nilai similaritasnya lebih besar dari threshold yang diberikan. Hasil dari langkah ini direpresentasikan oleh baris berikut ini:

$$\langle N_{SP}^i, N_{PP}^j, \tau_{label}, Sim_{label_\tau} \rangle \quad (\text{III.6})$$

di mana  $i = 1..i_{max}$ ,  $j = 1..j_{max}$ ;  $N_{SP}^i$ ,  $i_{max}$  adalah label dari konsep ke  $i$  dan jumlah label dalam super peer  $SP$ ;  $N_{PP}^j$ ,  $j_{max}$  berturut-turut adalah label dari konsep

ke  $j$  dan jumlah dari label dalam  $PP$ ;  $N_{SP}^i$  adalah label dari konsep ke  $i$  dalam common ontology yang dikelola oleh super peer  $SP$ ; Dalam tuple III.6 sebuah threshold  $\tau_{label}$  digunakan dalam kalkulasi  $Sim_{label\tau}$ . Nilai dari  $Sim_{label\tau} \in \{0, 1\}$  adalah 1 jika nilai yang dihasilkan dari fungsi pencocokan lebih besar dari threshold  $\tau_{label}$  dan 0 apabila sebaliknya.

Langkah terakhir adalah menentukan relasi antara kandidat peer yang didapat pada langkah 2. Hasil dari langkah ini diekspresikan oleh tupel III.4 di atas.

### 3.3.2.2 Similaritas Menggunakan Struktur Konsep

Pada bagian sebelumnya, pencocokan label diaplikasikan pada level konsep paling atas untuk menghitung ukuran similaritas. Hasil dari proses pencocokan tersebut dapat di perhalus (fine tuned) dengan mempertimbangkan baik struktur internal dan struktur eksternal dari konsep yang direpresentasikan oleh sebuah label. Berikut ini akan dibahas mengenai kalkulasi dengan menggunakan struktur tersebut.

**Menghitung Nilai Similaritas dengan Struktur Internal** Similaritas antara dua konsep dapat diperoleh dengan membandingkan *language* dan *real attribute* dan tidak hanya dari persamaan antara uraian komponen mereka, tetapi juga dari persamaan antar struktur dari grafik yang mewakili mereka. Similaritas dari struktur internal dapat diperoleh dengan menghitung banyaknya attribute yang serupa dibagi jumlah maksimal attribut dari sebuah kelas.  $IS = \text{similarattribute} / [\text{maksattributpada sebuah kelas}]$  [Rahm and Bernstein, ] dan [Giunchiglia and Zaihrayeu, ]

Struktur internal dari sebuah konsep mengandung atribut atau propertinya, yang dapat menjadi language related (lihat file OWL) atau entitas dunia nyata berbasis pada atribut. Properti language related adalah sebuah label seperti owl:cardinality dan rdfs:label. Penggunaan properti berbasis bahasa tergantung pada taksonomi dari sumber dan ontologi. Properti dunia nyata menggambarkan beberapa entitas seperti nama, alamat dan gaji dari seseorang.

Pernyataan similaritas berbasis struktur internal menggunakan properti keterhubungan real world. Untuk menghitung digunakan entitas similaritas dari struktur graf yang di usulkan oleh Thanh-Le et al dalam [Euzenat and Valtech, ] dan [Aspnes and Shah, ] dan didefinisikan dengan persamaan III.7 berikut, di mana graph dapat digantikan oleh *ICworld*.

$$Sim_{graph}(c_1, c_2) = \frac{|P_c|}{\max(|P(c_1)|, |P(c_2)|)} \quad (III.7)$$

di mana  $|P_c|$ ,  $|P(c_1)|$ ,  $|P(c_2)|$  berturut-turut adalah jumlah atribut similaritas, jumlah atribut  $c_1$ , dan jumlah atribut  $c_2$ .

Merepresentasikan Similaritas Label Berbasis pada Struktur Internal Tupel III.8 dan III.9 digunakan untuk menyatakan nilai similaritas ketika struktur internal dari konsep diperhatikan.

$$\langle N_{SP}^i, N_{PP}^j, Sim_{IC_{lang}}, Sim_{IC_{world}} \rangle \quad (III.8)$$

$$\langle N_{SP}^i, N_{PP}^j, \tau_{IC_{lang}}, Sim_{IC_{lang\tau}}, \tau_{IC_{world}}, Sim_{IC_{world\tau}} \rangle \quad (III.9)$$

di mana  $i = 1..i_{max}$ ,  $j = 1..j_{max}$ ;  $N_{SP}^i$  berturut-turut adalah label dari konsep ke  $i$  dalam common ontology yang dikelola oleh super peer  $SP$ ;  $i_{max}$  adalah jumlah dari label super peer;  $N_{PP}^j$  adalah label dari konsep lokal ke  $j$  dari peer;  $PP$ ;  $j_{max}$  adalah jumlah label dalam peer  $PP$ ;  $Sim_{IC_{lang}} \in [0, 1]$  merepresentasikan nilai similaritas menggunakan atribut bahasa ;.  $Sim_{IC_{world}}$  merepresentasikan nilai similaritas menggunakan atribut atribut dunia (world) nyata sebagai struktur internal. ; pada tuple III.9, threshold  $\tau_{IC_{lang}}$  dan  $\tau_{IC_{world}}$  digunakan dalam kalkulasi dari  $Sim_{IC_{lang\tau}}$  and  $Sim_{IC_{world\tau}}$ . nilai dari  $Sim_{IC_{lang\tau}} \in \{0, 1\}$  dan  $Sim_{IC_{world\tau}} \in \{0, 1\}$  adalah 1 jika nilai yang dihasilkan oleh fungsi match lebih besar dari threshold ( $\tau_{IC_{lang}}$  dan  $\tau_{IC_{world}}$ ) dan bernilai 0 apabila sebaliknya

**Menghitung Nilai Similaritas dengan Struktur Eksternal** Perbandingan eksternal struktur adalah melihat set dari upper-class. Secara sederhana untuk menghitung eksternal struktur dari dua kelas adalah dengan melihat berapa jumlah upper-class yang sama akan dibagi dengan jumlah upper-class terbesar dari sebuah kelas [Madhavan et al., ].  $ES = \text{similar upper-class} / [\text{maks upper-class pada sebuah kelas}]$ .

Struktur eksternal dari konsep bergantung pada posisinya dalam sebuah hirarki konsep. Hal itu dapat didefinisikan dalam dua cara. Yang pertama memasukkan node yang terdapat pada subpohon di mana konsep adalah root. Maka struktur eksternal dari konsep mengandung hanya sub kelas nya. Dalam kasus ini, struktur eksternal dibentuk dari seluruh konsep yang dimiliki oleh jalur yang dimulai dari root pada hirarki dan berakhir pada konsep.

Tergantung pada bagaimana struktur eksternal didefinisikan, dua metode dapat digu-

nakan untuk mengkalkulasi nilai similaritas. Sebagai contoh, metode pencocokan Cupid diusulkan dalam [Madhavan et al., ] mempertimbangkan daun atau data atomik dapat dicapai dari sebuah konsep untuk mengkalkulasi nilai similaritas. Sedangkan metode yang diusulkan dalam [Euzenat and Valtech, ] mendefinisikan jarak, yang dikatakan sebagai upward ctopic distance berbasis pada himpunan dari super class. Dinyatakan dengan similaritas pencocokan struktur menggunakan super class. Similaritas antara dua konsep dan dinyatakan oleh persamaan III.10:

$$\langle N_{SP}^i, N_{PP}^j, Sim_{EC_{set-sp}}, Sim_{EC_{leave}} \rangle \quad (III.10)$$

$$\langle N_{SP}^i, N_{PP}^j, \tau_{EC_{set-sp}}, Sim_{EC_{set-sp}\tau}, \tau_{EC_{leave}}, Sim_{EC_{leave}\tau} \rangle \quad (III.11)$$

di mana  $i = 1..i_{max}$ ,  $j = 1..j_{max}$ ;  $N_{SP}^i$  adalah label dari konsep ke i dalam common ontology yang dikelola oleh super peer  $SP$ ;  $i_{max}$  adalah jumlah dari label super peer;  $N_{PP}^j$  adalah label ke lokal konsep ke j dari peer;  $PP$ ;  $j_{max}$  adalah jumlah label dari peer  $PP$ ;  $Sim_{EC_{set-sp}} \in [0, 1]$  adalah bilangan nyata yang menspesifikasikan nilai similaritas dari struktur eksternal dari himpunan super class;  $Sim_{EC_{leave}} \in [0, 1]$  adalah bilangan nyata yang menspesifikasikan nilai similaritas dari eksternal struktur dari himpunan daun; Pada tuple III.9, threshold  $\tau_{EC_{set-sp}}$  dan  $\tau_{EC_{leave}}$  digunakan dalam kalkulasi dari  $Sim_{EC_{set-sp}\tau}$  dan  $Sim_{EC_{leave}\tau}$ . Nilai dari  $Sim_{EC_{set-sp}\tau} \in \{0, 1\}$  dan  $Sim_{EC_{leave}\tau} \in \{0, 1\}$  adalah 1 jika nilai yang dihasilkan oleh fungsi match lebih besar dari thershold ( $\tau_{EC_{set-sp}}$  dan  $\tau_{EC_{leave}}$ ) dan bernilai 0 untuk sebaliknya.

di mana H adalah hirarki konsep, dan ) berturut-turut adalah super kelas dan . Upward ctopic distance yang diusulkan Euzenat, mengekspresikan similaritas dari dua konsep sebagai rasio di antara sejumlah super kelas umum yang berelasi secara kuat dan sejumlah total super kelas dari kedua konsep. Sebagai catatan bahwa super kelas umum yang kuat berelasi ditentukan oleh perbandingan similaritas dari label dengan menggunakan WordNet.

Merepresentasikan Similaritas Label berbasis pada Struktur Eksternal Model III.12 dan III.13 merepresentasikan nilai similaritas ketika struktur eksternal dari konsep diper-timbangkan.

$$Sim_{EC_{set-sp}}(c_1, c_2) = \frac{|UC(c_1, H) \cap UC(c_2, H)|}{|UC(c_1, H) \cup UC(c_2, H)|} \quad (III.12)$$

di mana  $UC(c_1, H)$  adalah himpunan super class dari  $c_1$ , yang sampai level n ke paling atas. Hal ini mengacu pada upward ctopic distance [Euzenat et al., 2004], yang mem-

bandingkan himpunan similaritas dari super calss dibagi dengan bilangan maksimum dari himpunan super class dari kedua konsep.

$$Sim_{EC_{leave}}(c_1, c_2) = \frac{\left| \begin{array}{l} \{x \mid x \in leaves(c_1) \wedge \exists y \in leaves(c_2), stronglink(x, y)\} \\ \cup \{x \mid x \in leaves(c_2) \wedge \exists y \in leaves(c_1), stronglink(y, x)\} \end{array} \right|}{\left| leaves(s_1) \cup leaves(s_2) \right|} \quad (III.13)$$

di mana  $leaves(c_1)$  = himpunan daun dalam subpohon yang berakar pada  $c_1$ . Itu tidak menghitung sebuah 1-1 pencocokan bipartite karena hal itu sangat mahal perhitungannya dan akan mencegah pemetaan m:n.

### 3.4 Semantic Agreement

Konflik semantik timbul ketika dua sistem tidak menggunakan interpretasi yang sama terhadap suatu informasi. Bentuk paling sederhana dari ketidaksetujuan dalam menginterpretasikan informasi adalah homonim (menggunakan kata yang sama dengan arti yang berbeda), dan sinonim (menggunakan kata berbeda dengan arti yang sama). Dalam kasus ini, semantik dari informasi harus dipertimbangkan dalam rangka memutuskan bagaimana item informasi yang berbeda berhubungan satu dengan yang lainnya. Yaser [Bishr, 1997] telah membagi keragaman skematik ke dalam beberapa kelompok: Berbeda dalam kelas seperti sinonim, homonim, berbeda dalam atribut kelas, metode dan batasan integritas. Berbeda dalam atribut, seperti domain, unit, nilai default dan tipe data Berbeda dalam hirarki, seperti kelas, atribut, tingkat generalisasi dan agregasi.

Untuk mengatasi perbedaan semantik di antara konten dari peer, maka agreement dibuat untuk merepresentasikan hasil dari pencocokan atau pemetaan konsep dari satu peer ke konsep dari peer yang lain. Seperti yang sudah didiskusikan di atas, pencocokan konsep adalah kombinasi dari (1) analisis linguistik pada label konsep, (2) pencocokan label yang berdasarkan pada definisi dari nilai similaritas menggunakan struktur (internal dan eksternal) dari konsep dan (3) identifikasi dari relasi semantik di antara konsep. Dalam langkah yang berbeda dari pencocokan semantik, sebuah konsep dipandang sebagai hirarki dari label terminologikal yang berisi informasi struktur dan semantik. Pendekatan agreement dalam penulisan ini berdasarkan beberapa asumsi yang meliputi fakta bahwa bahasa yang sama digunakan untuk merepresentasikan skema ekspor dan ontologi yang terdapat dalam Agreement context dan konteks

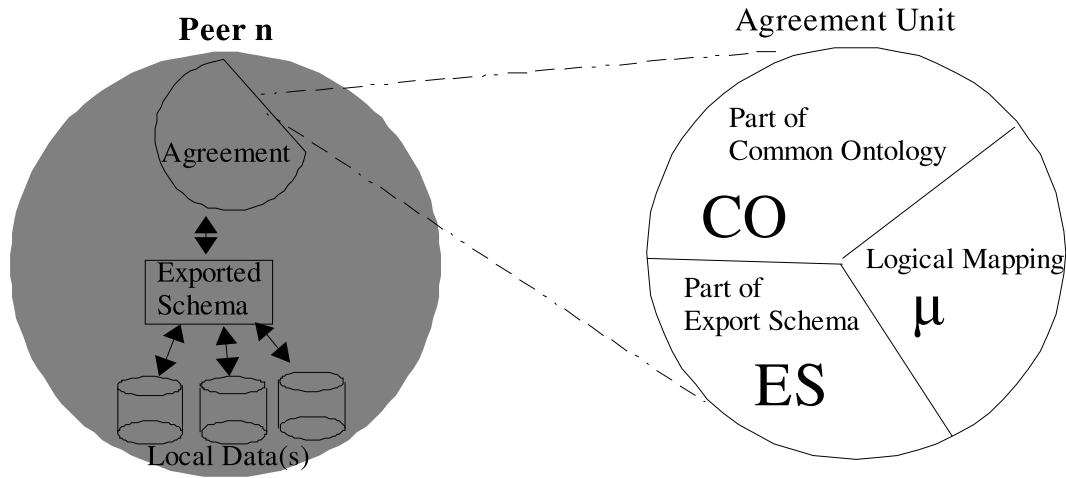
komunitas secara bersamaan, di mana label dari konsep tersebut mengandung beberapa informasi semantik dan pada akhirnya tidak ada individual ataupun instant yang berasosiasi dengan common ontology.

### 3.4.1 Unit Agreement

Fitur kunci dari pendekatan semantic agreement adalah unit agreement, yang berguna untuk mengekspresikan pemetaan semantik antara konsep yang diekspor oleh peer dan konsep ontologi berbagi (shared ontology) yang dikelola oleh super peer. Sebuah unit agreement berkorespondensi dengan interpretasi semantik atau view dari satu atau lebih konsep ontologi. Hal itu digunakan untuk mengadaptasi semantik dari konsep global ke view lokal dari peer. Hal tersebut dilakukan dengan menggunakan lokal sebagai view langkah integrasi semantik dalam hal (1) struktur atau bagian dari struktur dari konsep skema ekspor yang didefinisikan berdasarkan pada struktur dari konsep ontologi dan (2) konteks semantik dari konsep ontologi dikembangkan atau diselesaikan menggunakan predikat logik untuk mengkonfirmasi ke konteks lokal. Sebuah unit agreement yang dibuat oleh satu peer  $P_i$  dikomposisikan dari pemetaan Pemetaan ini dapat merupakan hubungan satu-ke-satu atau satu-ke-banyak yang mendefinisikan antara konsep lokal dan konsep ontologi. Relasi banyak-ke-banyak tidak dibahas dalam prototipe karena menyimpang dari skope dari disertasi ini. Seperti tergambar dalam gambar 3.4, sebuah unit agreement mengkapsulkan tiga komponen: (1) satu atau lebih konsep ontologi, (2) sebuah konsep skema ekspor, dan (3) fungsi pemetaan logik antara komponen pertama dan kedua. Ketiga komponen dari unit agreement direpresentasikan dengan deskripsi RDF/OWL dalam protipe pendekatan. Sebuah unit agreement merepresentasikan satu half agreement dalam arti untuk bertukar data, kedua peer  $P_i$  dan  $P_j$  perlu untuk (1) mendefinisikan unit agreement yang berkorespondensi antara konsep-konsepnya dan ontologinya dan (2) mengkombinasikan atau menggabungkan unit agreement yang berisi hanya setengah dari informasi korespondensi semantik yang dibutuhkan, untuk membuat full semantic agreement yang menghubungkan isi dari dua peer  $P_i$  dan  $P_j$ .

### 3.4.2 Pengembangan dari Unit Agreement

Gambar 3.4 mengilustrasikan urutan dari langkah-langkah untuk membuat unit agreement. Berikut ini ringkasannya, pembuatan unit agreement dengan memberikan deskripsi prosedural dari aspek utama dan langkah-langkah dari perhitungan dari pengukuran similaritas. Langkah 1: input berisi subset dari common ontology dan skema ekspor



Gambar 3.4: Representasi *Unit Agreement*)

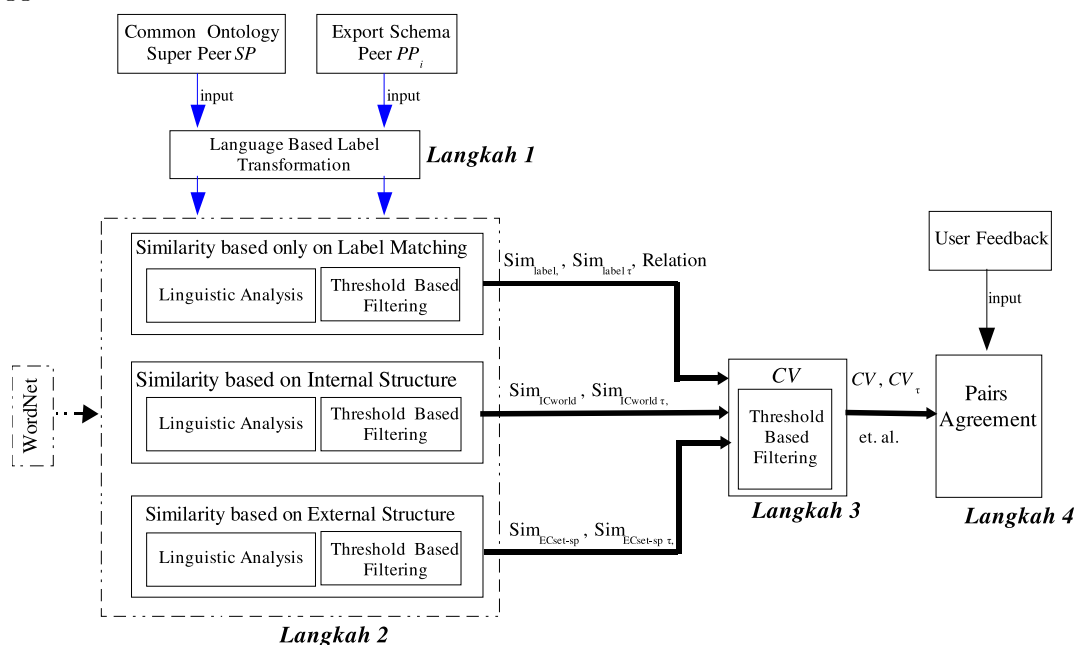
dari peer PP yang menginisiasikan proses pengembangan unit agreement. Sebagai tambahan untuk input, taksonomi terminologi WordNet digunakan. Langkah pertama digunakan untuk mentranformasi initial input untuk kalkulasi similaritas pada langkah berikutnya.

Langkah 2: melibatkan tiga tipe kalkulasi dari pengukuran similaritas, yaitu pencocokan label (label matching), similaritas berdasar internal (internal based similarity) dan similaritas berdasar eksternal (external based similarity). Setiap kombinasi dari ketiga metode digunakan untuk menghitung similaritas antara konsep. Hasil dari langkah ini adalah relasi awal antara label-label yang diekspresikan sebagai tupel. Kandidat pasangan konsep diputuskan untuk relasi semantik akhir yang didefinisikan pada langkah 3.

Langkah 3: mengkombinasikan hasil dari pencocokan label (label matching) dan similaritas struktur pada langkah 2 dengan mengasosiasikan ke setiap metode untuk mengkalkulasikan nilai kepercayaan (confident). Pengukuran similaritas akhir didefinisikan dengan  $CV = \sum \gamma_i * SimilarityMethods$ , dimana  $\sum \gamma_i = 1$ ,  $\gamma_i$  adalah nilai bobot dari setiap pendekatan perhitungan similaritas. Confidence value will be filtered by using threshold value ( $\tau_{CV}$ )

Langkah 4: mendefinisikan unit agreement. Feedback pengguna dibutuhkan untuk memutuskan setiap ke-ambiguan relasi yang tidak diklarifikasi oleh langkah sebelumnya. Beberapa nilai threshold sangat penting dalam penentuan pengukuran similaritas, kandidat pasangan konsep yang dapat dihubungkan dengan relasi, dan unit agreement. Tidak ada pendekatan yang tepat atau pasti untuk menemukan nilai threshold yang

benar/tepat. Dalam simulasi dan eksperimen yang telah dilakukan, digunakan metode try and error untuk menemukan nilai threshold. Maka, nilai threshold tidaklah generik, yang berarti bahwa nilai tersebut dapat saja berbeda untuk domain yang berbeda. Karena kesulitan untuk menemukan nilai threshold dan confident yang tepat, maka sangatlah penting untuk menentukan hasil awal atau relasi yang dipresentasikan kepada pengguna untuk keputusan akhir.



Gambar 3.5: Langkah dari Pengembangan *Unit Agreement*)

### 3.4.3 Pemrosesan Kueri Menggunakan Unit Agreement

Telah didiskusikan hal-hal yang berkaitan dengan proses atau urutan dari langkah-langkah yang harus dilakukan oleh peer-peer yang berbeda yang berpartisipasi dalam lingkungan berbagi informasi (information sharing). Satu asumsi penting yang dibuat yaitu satu common ontology tersedia dan disimpan dalam satu atau lebih super peer yang mempunyai sumber daya dan kapasitas untuk sebagian mengkoordinasi interaksi di antara peer. Sebagai tambahan, sebuah standar kosa kota atau taksonomi term seperti WordNet diperlukan untuk menyediakan referensi arti bagi label-label yang berasosiasi dengan konsep yang terdapat dalam common ontology. Satu ide paling penting dari penelitian dalam Wicaksana [Wicaksana, 2006], yang diambil dari penggunaan common ontology, yaitu penemuan korespondensi semantik di antara peer dilakukan dalam dua langkah: (1) ketika sebuah peer bergabung (join) dengan lingkungan mediasi yang dinamis, peer mempublikasikan sebagian dari kontennya dan menghubungkan (link)

konsepnya ke ontologi global melalui unit agreement yang menyimpan metadata dan fungsi transformasi yang digunakan untuk mencocokkan konsep. (2) ketika sebuah peer memerlukan data remote, peer terlebih dahulu harus mengirim sebuah permintaan untuk melokalisasikan peer yang relevan dan kemudian menggabungkan atau menghubungkan unit agreement-nya dengan satu atau lebih unit agreement dari peer yang relevan.

Strategi pemrosesan kueri dan protokol adalah sangat penting untuk agreement berbasis pendekatan mediasi [Wicaksana, 2006]. Pemrosesan kueri melibatkan satu langkah penulisan kembali kueri (query rewriting) yang menggunakan hasil dari pencocokkan konsep dan proses pembuatan agreement. Di situ ditekankan langkah utama dari proses tersebut, khususnya memfokuskan pada karakteristik semi otomatis dari dua langkah akhir yang dalam hal ini pengguna atau umpan balik ahli domain (domain expert) digunakan untuk memilih pasangan yang berhubungan berdasarkan pada kalkulasi dari nilai similaritas pada fase sebelumnya. Isu kueri utama adalah bagaimana menulis kembali (rewrite), mengirim kepada peer yang tepat dalam hal konsep terdapat dalam skema eksportnya. Isu lainnya adalah mengenai karakteristik dari model data atau sistem informasi semi terstruktur. Sistem informasi terdistribusi yang semi terstruktur dicirikan dengan kenyataan bahwa (1) skema komponen tidak dikenal sebelumnya, tetapi lebih pada implisit dalam data, (2) skema data implisit dapat saja menjadi relatif besar dan dapat berubah secara teratur, dan (3) data sama sekali tidak strongly typed, yang berarti bahwa atribut dengan nama yang sama mungkin saja berubah sebagaimana mereka digunakan oleh komponen yang berbeda.

Mengacu kepada analogi dan asumsi dari [Tempich et al., ], Tempich et al membandingkan pemrosesan kueri untuk aspek atau langkah yang diambil oleh manusia ketika dia mencari jawaban atas pertanyaan, maka pendekatan berbasis agreement untuk menangani kueri dijelaskan oleh langkah-langkah prosedur berikut ini: unit agreement dibuat dari konsep-konsep peer PPI sampai common ontology dalam super peer SP untuk mengkonfirmasi semantik-nya. kueri dikirimkan oleh peer dengan melihat, memperhatikan, menemukan peer data dan kemudian meminta dan menerima unit agreement dari peer yang terlokalisasi (located peer). respon kueri dapat berasal dari beberapa sumber data dan terintegrasi untuk membuat hasil akhir oleh peer yang menginisiasi kueri.

## 3.5 Peer Bergabung, Peer Meninggalkan dan Penemuan Peer

Pencocokan label dan konsep dan metode pengukuran similaritas yang telah disajikan di atas adalah digunakan untuk menangani isu keragaman. Aspek dinamis dari sistem ditandai oleh tingkah laku peer. Pemrosesan kueri bergantung pada dua jenis aktivitas peer: (1) peer bergabung dan meninggalkan dan (2) menemukan peer yang relevan dari kueri.

### 3.5.1 Peer Bergabung dan Meninggalkan

Peer diijinkan untuk bergabung dan meninggalkan framework dinamis secara bebas dan teratur (frequently). Pada saat memulai sistem, hanya ontologi domain berbagi dibuat dan dikelola oleh satu atau lebih super peer. Ketika sebuah super peer bergabung dengan sistem, super peer mendaftar dengan super peer pengelola yang ada saat itu sebagai back-up. Ketika terdapat lebih dari satu super peer, super peer tersebut diorganisasi dalam hirarki. Ketika sebuah peer bergabung ke sistem, peer mendaftar konten dari skema eksportnya dengan mengirimkan permintaan Publish ke super peer pengelola untuk membuat sebuah unit agreement. Pada penjelasan sebelumnya, telah dijelaskan metadata super peer dan tabel sistem informasi yang perlu untuk di-update mengikuti penerimaan dari permintaan Publish. Pada sisi super peer, sebuah pesan Publish digunakan untuk menyimpan status dari peer yang aktif dan untuk menspesifikasikan konsep dan peer di mana unit agreement dibuat. Pada sisi peer, mengirim informasi bookkeeping dan status kepada super peer dan melakukan langkah-langkah untuk membuat unit agreement antara konsep skema eksport dan konsep ontologi. Untuk meninggalkan sistem, sebuah super peer harus menginisiasi satu langkah sinkronisasi dengan super peer yang tersedia selanjutnya dalam hirarki untuk mentransfer seluruh metadata bookeping dan ontologi. Peer yang lain meninggalkan sistem dengan cara sangat sederhana yaitu mengirim pesan status *NOT ACTIVE* ke super peer yang ada/aktif.

### 3.5.2 Penemuan Peer

Peer bergabung dan meninggalkan dapat mempengaruhi ketersediaan dan berbagi data. Ketika sebuah kueri dikirim oleh sebuah peer, beberapa operasi/aksi dilakukan. Pertama, informasi status yang disimpan oleh super peer selama pemrosesan pesan Publish digunakan untuk menentukan peer yang aktif. Selanjutnya, simpanan metadata unit

agreement digunakan untuk menemukan peer kandidat dengan konten yang cocok dengan konsep dari kueri. Peer kandidat yang terpilih mengkalkulasi similaritas semantik (matchmaking) antara unit agreement-nya dari peer kueri (peer yang pada awal mengirimkan kueri). Sebuah agreement adalah sebuah baris (tuple) yang mengandung (1) sebuah konsep dari common ontology, (2) sebuah konsep dari skema eksport, dan (3) pemetaan untuk mengkonversi di antara common ontology dan skema eksport. Untuk mencocokkan unit agreement dari sebuah peer, satu kandidat peer membandingkan konsep ontologi yang terlibat dengan menggunakan relasi yang telah didefinisikan dalam Wicaksana. Unit agreement adalah terkonfirmasi jika konsep ontologi-nya ekuivalen, serupa atau dihubungkan oleh relasi superclass dan subclass.

### 3.6 Isu Perubahan Versi Ontologi dan Skema Lokal

Ontologi adalah suatu cara yang berarti bagi sebuah agreement pada konotasi dari *things* di antara kelompok yang berinteraksi, baik itu manusia maupun komputer. Dengan mempercayakan kepada sebuah ontologi, sebuah kelompok interaksi mendeklarasikan bahwa dia peduli terhadap arti dari kosa kata dalam ontologi dan bahwa dia akan membagi arti tersebut selama interaksinya, yang menjamin konsistensi selama interaksi. Untuk komit pada sebuah ontologi tidak harus memerlukan perubahan apapun dalam lingkungan kerja dari kelompok interaksi. Karena ontologi dapat digunakan untuk membagi konseptualisasi antara kelompok komunikasi, kegunaannya secara teoritikal bermanfaat di mana sebuah agreement dalam arti *things* penting.

Pertama-tama, sebuah ontologi menyediakan kelompok interaksi dengan kosa kata umum (common) dan standar dengan cara ini menggunakan terminologi. Lebih jauh, pengetahuan yang tiap kelompok interaksi biasanya menggunakan secara implisit, dapat dibuat eksplisit dengan ontologi yang memungkinkan untuk membagi dan re-use pengetahuan tersebut.

Dunia ini secara teratur berubah, hal tersebut mempengaruhi yang terdapat di dalamnya. Pada sebagian aplikasi, ontologi tidaklah statis. Selain itu, ontologi harus beradaptasi dengan perubahan dari domain aplikasi dan perluasan area. Sehingga evolusi ontologi adalah salah satu aspek dalam pemeliharaan ontologi [Borden, 2007].

Perubahan dalam konseptualisasi membutuhkan modifikasi dari ontologi. Evolusi ontologi menyebabkan masalah kemampuan pengoperasian, di mana akan menghambat re-use yang efektif. Dukungan untuk menangani perubahan tersebut dibutuhkan. Hal

ini khususnya penting dalam sebuah lingkungan yang terdesentralisasi dan tidak terkontrol, seperti web, di mana perubahan terjadi tanpa penyesuaian / standarisasi. Lebih lagi dalam sebuah lingkungan yang terkontrol, hal itu dapat menghasilkan hasil yang tidak diharapkan dan tidak diketahui. Dengan munculnya Semantic Web, perubahan yang tidak terkontrol tersebut bahkan akan mempunyai lebih banyak pengaruh / benturan, karena komputer akan menggunakan data. Tidak ada lagi manusia dalam rantai tersebut yang menggunakan sejumlah besar pengetahuan latar belakang dan *heuristik implisit* dapat menandai kombinasi kesalahan disebabkan oleh perubahan yang tidak terduga / tidak diharapkan. Masalah ini lebih buruk, karena terdapat sejumlah besar ketergantungan antara sumber data, aplikasi dan ontologi. Perubahan sampai nanti akan mempunyai efek samping yang jauh dari pencapaian.

Seringkali tidak mungkin untuk mensinkronisasikan perubahan pada sebuah ontologi dengan modifikasi pada aplikasi dan sumber data yang menggunakannya. Maka, sebuah metode versioning dibutuhkan untuk menangani revisi dari ontologi dan akibatnya pada sumber yang ada.

### 3.6.1 Manajemen Perubahan Versi Ontologi

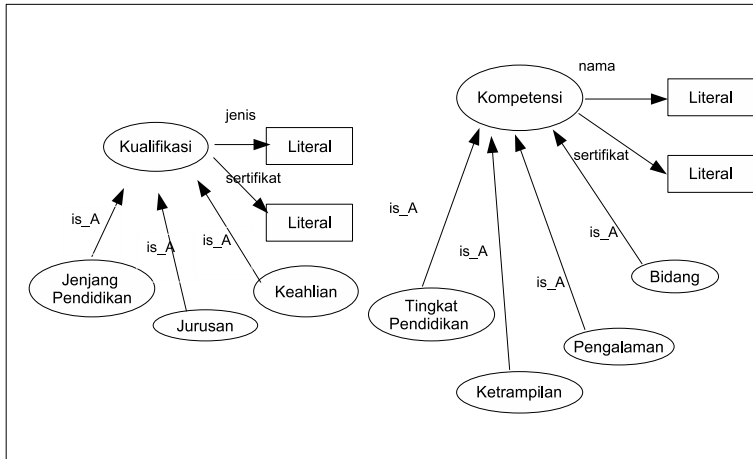
Salah satu tool yang dapat digunakan untuk membandingkan dua ontologi berdasarkan versinya adalah PROMPT TAB, yang merupakan salah satu plug-in dalam Protégé. Dengan menggunakan Prompt [Klein and Noy, 2003], kita dapat: membandingkan dua versi dari ontologi yang sama dan membuat sebuah gabungan versi menghapus frame dari project menggabungkan dua ontologi menjadi satu mengekstraksi bagian dari sebuah ontologi Membandingkan Dua Versi Ontologi yang sama Mode ini memungkinkan kita untuk membandingkan dua versi yang berbeda dari ontologi yang sama. Prompt menggunakan satu himpunan heuristic untuk membandingkan dua versi ontologi dari sebuah ontologi dan menemukan frame yang berkorespondensi pada dua versi itu. Untuk membandingkan nama dan tipe dari frame, Prompt melihat struktur dari dua versi, sebagai saran misalnya, yaitu sebuah unmatched sibling tunggal dari parent yang sama dalam dua versi dapat menjadi frame yang sama dengan nama yang berbeda. Dengan berkonsentrasi pada perbedaan di antara dua project yang serupa (daripada similaritas dari dua project yang berbeda ketika digabung), membandingkan dapat memberi hasil lebih baik bagi penyesuaian yang dibutuhkan dalam mengontrol versi.

### 3.7 Ilustrasi Contoh

Dengan meminjam ide contoh berikut dari beberapa paper diambil dari [Bourse et al., 2002], [Bizer et al., 2005] dan [Wicaksana and Guritno, 2005], maka akan diilustrasikan penerapan layanan job matching yang terdapat di Indonesia. Berikut ini ilustrasinya: suatu perusahaan (penyedia lapangan kerja) membutuhkan calon pegawai baru dengan kualifikasi tertentu. Seperti diketahui di negara kita lulusan dari sebuah perguruan tinggi biasanya diberikan label lulusan dengan jenjang dan bidang studi jurusan tertentu. Sehingga para penyedia lapangan kerja saat membutuhkan atau mencari calon pegawai baru (pencari kerja) maka penyedia lapangan kerja tersebut akan menuliskan kualifikasi, yaitu jenjang pendidikan (misalnya Strata 1/S1, ataupun Diploma 3 / D3) dan bidang studi atau jurusan yang dibutuhkan, serta umumnya menuliskan kebutuhan khusus yang diharapkan (misalnya terampil bahasa pemrograman C) . Sedangkan dari sisi calon pegawai (pencari kerja) biasanya secara umum akan menuliskan kualifikasi dan juga ketrampilan yang dimilikinya (misalnya dapat melakukan pemrograman menggunakan bahasa pemrograman C dan VB). Dari ilustrasi di atas, dapat disimpulkan bahwa antara organisasi satu dengan organisasi lainya dapat saja mempunyai *view* yang berbeda terhadap suatu kebutuhan yang sama.

Untuk itu, diperlukan suatu pemecahan untuk menjembatani keberbedaan tersebut. Gambar 3.6 menggambarkan view yang berbeda terhadap kebutuhan yang sama dalam domain layanan job matching. Mengacu kepada pendekatan semantic agreement [Wicaksana, 2006] yang dapat digunakan untuk mengatasi keragaman semantik sehingga dapat menjembatani gap antara konten yang terdapat pada dua peer. Kunci utama dari pendekatan tersebut adalah pengukuran similaritas semantik yang berbasis pada perbandingan analisis linguistik dari struktur konsep (internal dan eksternal).

Pendekatan tersebut diusulkan dalam suatu lingkungan P2P, yang dinamis dan otonom. Dalam lingkungan yang dinamis dan otonom, administrator harus secara terus menerus memonitor common ontology. Monitoring secara manual adalah sangat mahal dan tidak akan dapat ter-skala dengan baik apabila jumlah peer meningkat. Untuk itu sangat penting mengembangkan teknik yang secara otomatis memonitor dan melakukan usulan perbaikan agreement yang telah dibentuk. Untuk dapat melakukan pemeliharaan agreement, beberapa langkah berikut ini harus dijalankan: 1. Menghitung perubahan yang terjadi pada skema lokal dan ontologi 2. Melakukan / mencari algoritma untuk perubahan 3. Menggunakan dan mengguji perubahan. Berikut adalah gambar dari



Gambar 3.6: Skema Ekspor dari Peer PP1 dan PP2 untuk contoh layanan Job Matching

proses pemeliharaan agreement, dan bagian mana yang menjadi titik berat riset ini.

Penelitian ini hanya dititikberatkan pada bagaimana menghitung perbedaan yang terjadi agar dapat ditentukan usulan algoritma yang digunakan, apakah menggunakan algoritma kompleks atau algoritma sederhana. Sehingga penelitian ini akan menghasilkan bagaimana mengukur perubahan dan menentukan algoritma mana yang digunakan.

Yang dimaksud dengan algoritma kompleks adalah algoritma yang terdapat pada pendekatan semantic agreement yang dikembangkan oleh Wicaksana [Wicaksana, 2006] yaitu menghitung agreement dengan langkah-langkah sebagai berikut:

1. Label Matching
2. Struktural Internal
3. Struktural Eksternal

Uraian detail dapat dilihat pada acuan [Wicaksana, 2006] yang diringkas dalam algoritma sebagai di bawah ini :

### Menghitung Perubahan

---

**Algorithm 1** Half Agreement at Class level

---

**Input:**

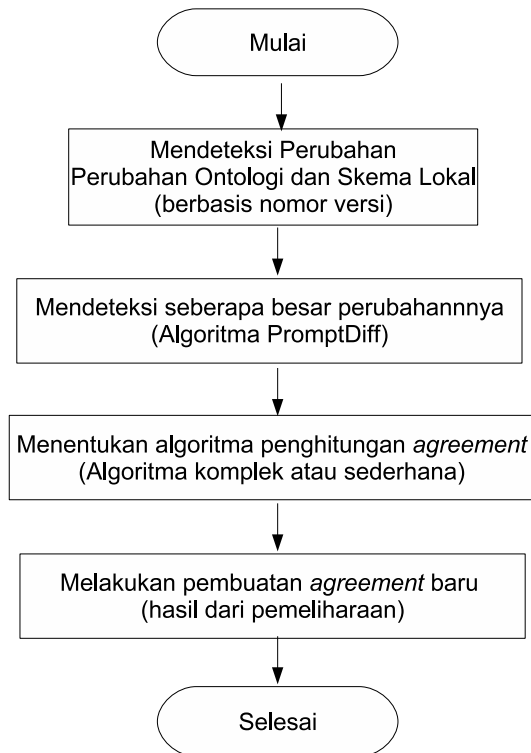
- 1: Common Ontology  $(N, A, l)$
- 2: Local Schema  $(N', A', l')$
- 3: WordNet
- 4: Threshold value  $(\tau_{label}, \tau_{IC_{lang}}, \tau_{IC_{world}}, \tau_{EC_{set-sp}}, \tau_{EC_{leave}}, ThCv)$
- 5: Weighting value  $\gamma_x$

**Steps:**

- 1: **for**  $k = 1$  to  $o$  **do**
- 2:   **for**  $l = 1$  to  $p$  **do**
- 3:     Language Process:  $p_{l_{c_k}}$  and  $p_{k_{c_l}}$
- 4:   **end for**
- 5: **end for**
- 6: **for**  $i = 1$  to  $m$  **do**
- 7:   **for**  $j = 1$  to  $n$  **do**
- 8:     Calculate:  $Sim_{label}(c_i, c_j)$ ,  $LCS(c_i, c_j)$  and  $Sim_{label\tau}(c_i, c_j)$  {Label Matching}
- 9:     Calculate:  $Sim_{IC_{lang}}(c_i, c_j)$ ;  $Sim_{IC_{lang\tau}}(c_i, c_j)$ ,  $Sim_{IC_{world}}(c_i, c_j)$  and  $Sim_{IC_{world\tau}}(c_i, c_j)$  {Internal Structure Matching}
- 10:    Calculate:  $Sim_{EC_{set-sp}}(c_i, c_j)$ ;  $Sim_{EC_{set-sp\tau}}(c_i, c_j)$ ,  $Sim_{EC_{leave}}(c_i, c_j)$  and  $Sim_{EC_{leave\tau}}(c_i, c_j)$  {External Structure Matching}
- 11:    Calculate:  $CV$
- 12:   **end for**
- 13: **end for**

**Output:**

- 1:  $\langle C_{SP}^i, C_{PP}^j, Sim_{label}(c_i, c_j) \rangle$  and  $\langle C_{SP}^i, Rel, C_{PP}^j, \tau_{label}, Sim_{label\tau}(c_i, c_j) \rangle$
  - 2:  $\langle C_{SP}^i, C_{PP}^j, Sim_{IC_{lang}}(c_i, c_j), Sim_{IC_{world}}(c_i, c_j) \rangle$  and  $\langle C_{SP}^i, C_{PP}^j, \tau_{IC_{lang}}, Sim_{IC_{lang\tau}}(c_i, c_j), \tau_{IC_{world}}, Sim_{IC_{world\tau}}(c_i, c_j) \rangle$
  - 3:  $\langle C_{SP}^i, C_{PP}^j, Sim_{EC_{set-sp}}(c_i, c_j), Sim_{EC_{leave}}(c_i, c_j) \rangle$  and  $\langle C_{SP}^i, C_{PP}^j, \tau_{EC_{set-sp}}, Sim_{EC_{set-sp\tau}}(c_i, c_j), \tau_{EC_{leave}}, Sim_{EC_{leave\tau}}(c_i, c_j) \rangle$
  - 4:  $CV(c_i, c_j)$  {final result will adjust with user feedback}
  - 5: Main output is candidate half agreement.
  - 6: Next step, consider relation in candidate half agreement.
-



Gambar 3.7: Alur Pemeliharaan Agreement

Berikut ini adalah langkah langkah dalam menghitung perubahan skema dan ontologi sehingga akan didapatkan sejumlah hasil yang dapat digunakan untuk memberikan usulan pertimbangan pemeliharaan agreement. Sebagai catatan untuk mengingatkan kembali bahwa dalam penelitian ini, digunakan nomor versi (mengacu ide dari DNS table) dari ontologi sebagai identifikasi dari ontologi yang sama yang dapat saja berubah kontennya, sehingga akan memunculkan versi berbeda dari ontologi yang sama. 1. Mendeteksi apakah terdapat perubahan dalam ontologi atau lokal skema 2. Membandingkan dua versi ontologi 3. Melakukan kuantifikasi / pembobotan terhadap perubahan yang ada 4. Menganalisis hasil perubahan

Mendeteksi Perubahan Ontologi Untuk mengetahui apakah terjadi perubahan, maka digunakan suatu metode yang berdasar kan pendekatan yang telah dikembangkan oleh Klein [Klein, 2002]

Membandingkan Dua Versi Ontologi Sebagian dari common ontology dan skema lokal diasumsikan berubah maka untuk mengetahui perbedaan antara satu versi dengan versi yang lain pada ontologi yang sama akan dilakukan perbandingan (*compare*). Metode yang digunakan untuk perbandingan ke dua versi tersebut adalah algoritma PromptDiff

[Noy, 2002].

Algoritma PromptDiff terdiri dari dua bagian: (1) sebuah himpunan yang diperluas dari heuristic matcher dan (2) algoritma fixed-point untuk mengkombinasikan hasil dari matcher untuk menghasilkan sebuah structural diff di antara dua versi. Tiap matcher mempekerjakan sejumlah kecil properti yang terstruktur dari ontologi untuk menghasilkan kecocokan. Langkah fixed-point memanggil matcher berulang kali, memasukan hasil dari satu matcher ke matcher lain, sampai langkah tersebut menghasilkan tidak adanya perubahan dalam diff. PromptDiff bekerja dengan membandingkan struktur dari versi ontologi.

Pendekatan mengotomatisasikan perbandingan berdasarkan pada dua observasi: [Noy, 2002] (1) ketika membandingkan dua versi dari ontologi yang sama, sejumlah besar besar fraksi dari frame tetap tidak berubah dan dua frame mempunyai tipe yang sama (misal, keduanya sama sama kelas, atau slot, dll) dan mempunyai nama yang sangat serupa, salah satu hampir pasti sebuah kopi atau gambaran dari yang lain. Kedua observasi tersebut adalah tidak benar apabila membandingkan dua ontologi yang berbeda yang datang dari sumber yang berbeda daripada dua versi yang sama dari ontologi. Mempertimbangkan sebuah kelas Universitas, misalnya. Pada dua ontologi yang berbeda, kelas dapat merepresentasikan baik itu kampus universitas, atau universitas sebagai sebuah organisasi, dengan departmen, fakultas nya dan lain sebagainya. Ketika menemukan kelas Universitas dalam dua versi dari ontologi yang sama, hal itu hampir dapat dipastikan bahwa itu pasti merepresentasikan konsep yang sama

Pada saat yang bersamaan, tugas dari perbandingan ontologi yang berbeda (sebagai contoh, untuk tujuan ontology merging atau integration) dan perbandingan versi dari ontologi yang sama adalah sangatlah dekat. Untuk kedua kasus, kita mempunyai dua ontologi yang overlapping dan perlu untuk menentukan sebuah mapping antara elemen-elemennya. Ketika membandingkan ontologi dari dari sumber yang berbeda, kita berkonsentrasi pada similaritas, sedangkan pada perbandingan versi, kita perlu untuk menggarisbawahi dan menekankan pada perbedaan, yang dapat menjadi proses yang melengkapi. Di sini digunakan heuristic untuk menyediakan saran dalam interactive ontology merging [Noy dan Musen, 2000]. Bagaimanapun, karena dalam PromptDiff setuju dengan dua versi dari ontologi yang sama, dapat lebih pasti dan yakin mengenai hasil dari heuristic hasilkan dan memerlukan input yang sangat jelas lebih sedikit dan verifikasi dari pengguna.

Proses perbandingan versi dari ontologi akan lebih sangat sederhana jika kita mempu-

nyai log dari perubahan versi-versi tersebut. Bagaimanapun juga, kecenderungan pada lingkungan desentralisasi dari pengembangan ontologi saat ini, adalah hal yang tidak realistis mengharapkan log seperti itu dapat tersedia. Sebagian besar tool pengembangan ontologi tidak menyediakan kemampuan logging. Library ontologi disiapkan untuk mempublikasikan versi dari ontologi tetapi tidak mencatat (log) perubahan. Oleh karena itu, kita dapat mengharapkan bahwa perlunya untuk membandingkan versi ketika catatan perubahan di antara versi tersebut tidak tersedia akan terus menerus meningkat.

Secara ringkas fitur dari PromptDiff: mendefinisikan pemikiran dari struktural diff di antara versi ontologi, untuk menemukan structural diff secara otomatis digunakan sebuah himpunan dari heuristic matcher, menjalankan algoritma fixed-point yang diperluas, yang mengkombinasikan matcher untuk menghasilkan diff.

### **Tabel Structural diff dan PromptDiff**

Didefinisikan structural diff antara dua versi ontologi.

#### **Difinisi III.1** (*Structural diff*)

*diberikan dua versi sebuah ontologi  $O$ ,  $V1$  dan  $V2$ , sebuah structural diff antara  $V1$  dan  $V2$ ,  $D(V1$  dan  $V2)$  adalah sebuah himpunan dari pasangan frame  $(F1, F2)$  di mana:  $F1 \in V1$  atau  $F1 = null$ ;  $F2 \in V2$  atau  $F2 = null$ ,  $F2$  adalah sebuah image dari  $F1$  (*matches*  $F1$ ), yaitu,  $F1$  menjadi  $F2$ . Jika  $F1$  atau  $F2$  adalah  $null$ , maka kita dapat mengatakan bahwa berturut-turut  $F2$  atau  $F1$  tidak cocok. Setiap frame dari  $V1$  dan  $V2$  muncul pada paling sedikit satu pasang. Untuk tiap frame  $F1$ , jika terdapat paling sedikit satu pasang yang mengandung  $F1$ , di mana  $F2$  tidak sama dengan  $null$ , maka tidak ada pasangan yang mengandung  $F1$  di mana  $F2 = null$  (jika kita temukan paling sedikit satu yang cocok untuk kita tidak perlu pasangan yang mengatakan bahwa  $F1$  adalah tidak cocok). Keadaan serupa adalah benar untuk  $F2$ .*

Catatan bahwa definisi mengimplikasikan bahwa untuk setiap frame  $F1$  dan  $F2$ , terdapat paling banyak satu entry  $(F1, F2)$ . Structure diff menggambarkan frame mana yang telah berubah dari satu versi ke versi lain. Bagaimanapun juga, untuk sebuah diff agar lebih berguna bagi pengguna, harus mengandung tidak hanya apa yang telah diubah tetapi juga beberapa informasi pada bagaimana frame berubah. Sebuah tabel PromptDiff yang merupakan hasil dari algoritma PromptDiff, menyediakan hal tersebut informasi lebih detail.

**Definisi III.2** (*Tabel PromptDiff*)

Diberikan dua versi ontologi  $O$ ,  $V1$  dan  $V2$ , sebuah tabel *PromptDiff* adalah himpunan tuple  $((F1, F2, renamevalue, operationvalue, mappinglevel)$  di mana: Terdapat tuple  $(F1, F2, renamevalue, operationvalue, mappinglevel)$  dalam tabel iff terdapat sepasang  $(F1, F2)$  pada structural diff  $D(V1, V2)$  *renamevalue* adalah benar jika nama frame untuk  $F1$  dan  $F2$  adalah sama; *renamevalue* adalah salah untuk sebaliknya. *operationvalue*  $\in OpS$ , di mana  $OpS = add, delete, split, merge, map$  *mappinglevel*  $\in MapS$ , di mana  $MapS = unchanged, isomorphic, changed$

Heuristic Matchers PromptDiff Algoritma PromptDiff mengkombinasikan sejumlah acak heuristic matcher, tiap tiap dari padanya melihat properti tertentu pada frame yang tidak cocok. Definisi 3 (Prinsip Monotonicity).  $M$  menjadi algoritma pencocokan dan  $T1$  dan  $T2$  menjadi tabel PromptDiff sebelum dan sesudah eksekusi dari  $M$ . Kemudian untuk setiap dua frame  $F1, F2$ , sehingga  $F1 \in V1$  dan  $F2 \in V2$ , jika pasangan  $(F1, F2)$  terdapat di dalam  $T1$  maka  $(F1, F2)$  terdapat di dalam  $T2$

Heuristic matcher membandingkan dua versi ontologi melihat situasi berikut ini:

1. Frame dengan tipe yang sama dengan nama yang sama. Secara umum, jika  $F1 \in V1$  dan  $F2 \in V2$  dan  $F1$  dan  $F2$  mempunyai nama dan tipe yang sama, maka  $F1$  dan  $F2$  cocok. Frame dapat bertipe class, slot, facet, atau instan.
2. Single unmatched sibling. Secara umum, jika  $C1 \in V1$  dan  $C2 \in V2$ ,  $C1$  dan  $C2$  cocok dan setiap dari kelas mempunyai unmatched subclass,  $subC1$  dan  $subC2$ , maka  $subC1$  dan  $subC2$  cocok. Juga terdapat matcher yang serupa untuk multiple unmatched sibling yang dapat dibedakan oleh himpunan slot nya.
3. Sibling dengan suffix atau prefix yang sama. Secara umum, jika  $C1 \in V1$  dan  $C2 \in V2$ , dan  $C1$  dan  $C2$  cocok, dan nama dari seluruh subclass dari  $C2$  kecuali untuk suffix yang konstan, maka subclass cocok.
4. Single unmatched slot. Secara umum, jika  $S1 \in V1$  dan  $S2 \in V2$ , dan  $S1$  dan  $S2$  cocok,  $invS1$  dan  $invS2$  berturut-turut adalah slot inverse untuk  $S1$  dan  $S2$ ,  $invS1$  dan  $invS2$  tidak cocok, maka  $invS1$  dan  $invS2$  cocok.
5. plil classes. Secara umum, jika  $C0 \in V1$ ,  $C1 \in V2$  dan  $C2 \in V2$ , dan setiap instance dari  $C0$ , image nya adalah sebuah instance baik dari  $C1$  atau  $C2$ , maka  $C0$ , adalah di belah (split) menjadi  $C1$  dan  $C2$ . Matcher serupa mengidentifikasi kelas kelas yang digabung.

Algoritma PromptDiff Noy dan Musen mengkombinasikan seluruh heuristic matcher dalam algoritma PromptDiff, sebuah algoritma fixed-point yang menghasilkan tabel PromptDiff yang lengkap bagi dua versi ontologi. PromptDiff menjalankan seluruh matcher sampai algoritma tersebut tidak menghasilkan perubahan yang baru dalam tabel. Karena tidak ada matcher yang menarik kembali hasil dari matcher sebelumnya atau hasilnya sendiri dari eksekusi sebelumnya (prinsip monotonicity), algoritma selalu konvergen (bertemu).

Algoritma yang digunakan pada pendekatan ini adalah akan melihat hasil algoritma perhitungan perbedaan jika melewati nilai tertentu akan dilakukan dengan algoritma label 3 jika sebaliknya akan digunakan algoritma 1 dari [Wicaksana, 2006].

---

**Algorithm 2** Calculate Different Version of Schema or Ontology

---

**Input:**

- 1: Common Ontology<sub>*x*</sub> ( $N, A, l$ )
- 2: Common Ontology<sub>*x-1*</sub> ( $N, A, l$ )
- 3: Local Schema<sub>*y*</sub> ( $N', A', l'$ )
- 4: Local Schema<sub>*y-1*</sub> ( $N', A', l'$ )
- 5: WordNet

**Steps:**

- 1: **for**  $k = 1$  to  $o$  **do**
- 2:   **for**  $l = 1$  to  $p$  **do**
- 3:     PromptDiff Process:  $p_{l_{c_k}}$  and  $p_{k_{c_l}}$
- 4:   **end for**
- 5: **end for**
- 6: **for**  $i = 1$  to  $m$  **do**
- 7:   **for**  $j = 1$  to  $n$  **do**
- 8:     Calculate: Different between Schema or Ontology {PromptTab}
- 9:   **end for**
- 10: **end for**

**Output:**

- 1: Total value of different
  - 2: Bandingkan dengan threshold value untuk menentukan perhitungan agreement
  - 3: Next step, memperhitungkan apakah ada perubahan, kalau ada perubahan pakai algoritma yang sesuai.
- 

Karena skema lokal yang dipergunakan dalam ilustrasi contoh dan pada eksperimen adalah dalam bahasa Indonesia dan WordNet adalah basis data leksikal bahasa Inggris, maka setiap konsep yang terdapat dalam skema lokal tersebut harus diterjemahkan ke dalam bahasa Inggris. Langkah dalam melakukan pemilihan kamus mengambil cara dan ide dalam penelitian [Banowosari and Wicaksana, 2007], pada gambar berikut ini:

---

**Algorithm 3** Simple Half Agreement at Class level

---

**Input:**

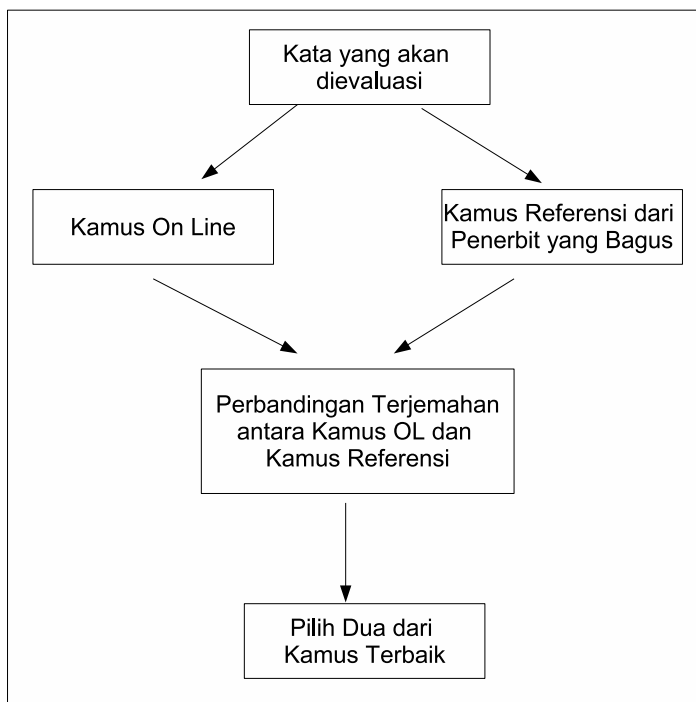
- 1: Common Ontology  $(N, A, l)$
- 2: Local Schema  $(N', A', l')$
- 3: WordNet
- 4: Threshold value  $(\tau_{label}, \tau_{IC_{lang}}, \tau_{IC_{world}}, \tau_{EC_{set-sp}}, \tau_{EC_{leave}}, ThCv)$

**Steps:**

- 1: **for**  $k = 1$  to  $o$  **do**
- 2:   **for**  $l = 1$  to  $p$  **do**
- 3:     Language Process:  $pl_{c_k}$  and  $pk_{c_l}$
- 4:   **end for**
- 5: **end for**
- 6: **for**  $i = 1$  to  $m$  **do**
- 7:   **for**  $j = 1$  to  $n$  **do**
- 8:     Calculate:  $Sim_{label}(c_i, c_j)$ ,  $LCS(c_i, c_j)$  and  
       $Sim_{label_\tau}(c_i, c_j)$  {Label Matching}
- 9:   **end for**
- 10: **end for**

**Output:**

- 1:  $\langle C_{SP}^i, C_{PP}^j, Sim_{label}(c_i, c_j) \rangle$  and  
    $\langle C_{SP}^i, Rel, C_{PP}^j, \tau_{label}, Sim_{label_\tau}(c_i, c_j) \rangle$
  - 2: Main output adalah kandidat agreement setelah melihat perubahan versi.
  - 3: Next step, memperhitungkan relasi dalam half agreement.
-

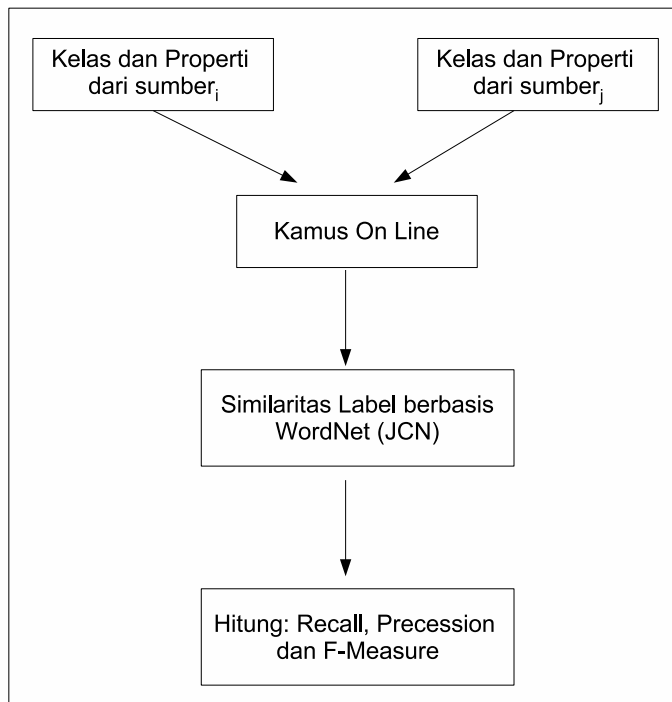


Gambar 3.8: Proses Pemilihan Kamus On-line)

[Banowosari and Wicaksana, 2007]

Pada contoh yang sudah diilustrasikan pada sub bab ilustrasi contoh di muka, berikut ini akan dijelaskan sebenarnya apa fungsi dari agreement, di mana agreement tersebut disimpan, mengapa agreement perlu dipelihara (maintain), dan bagaimana langkah pemeliharaan tersebut.

Seperti terlihat pada gambar III.6 bahwa terdapat dua skema lokal yang berberbeda, pada peer PP1, didefinisikan bahwa kelas Jenjang Pendidikan, Jurusan dan Keahlian adalah termasuk ke dalam kelas Kualifikasi, serta terdapat properti jenis dan sertifikat pada lokal skema peer PP1. Sedangkan pada lokal skema peer PP2 terdapat kelas Tingkat Pendidikan, Ketrampilan, Pengalaman, Bidang serta properti sertifikat dan masaBerlaku. Karena antara satu peer dengan peer yang lain mendefinisikan skema yang berbeda-beda maka akan timbul masalah pada saat pertukaran informasi antar keduanya. Di sinilah peran agreement, menjembatani gap antara satu sumber-sumber informasi yang berbeda. Mengacu kembali dalam [Staab, 2006], mengatasi keragaman informasi, beberapa pendekatan telah dilakukan, salah satunya adalah dengan menggunakan pendekatan interoperabilitas semantik yang digabungkan dengan P2P. Sebagai pivot dari pertukaran informasi antar kedua peer tersebut, maka model dari Wicaksana



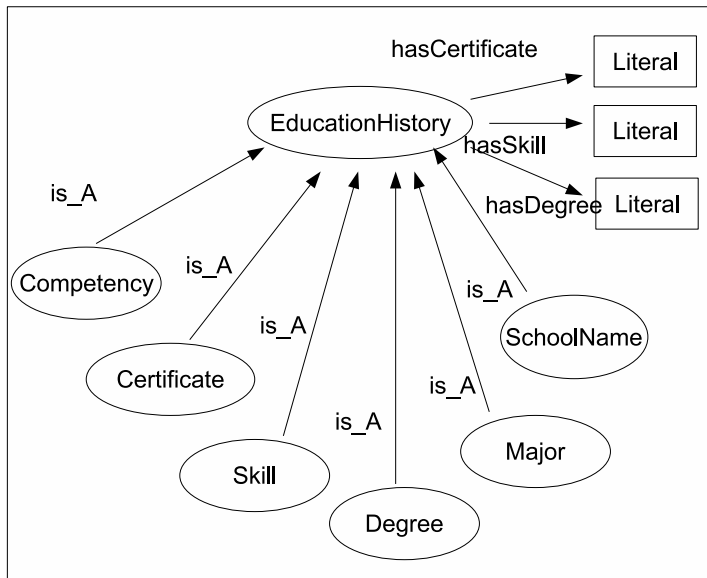
Gambar 3.9: Evaluasi Pencocokan Label Pada Bahasa Indonesia [Banowosari and Wicaksana, 2007]

diadopsi, yaitu membuat common ontology (CO) yang diletakan di Super Peer (SP), seperti yang terdapat pada gambar III.1

*Common ontology* yang digunakan dalam ilustrasi contoh adalah HR-XML, yaitu *Human Resources* ontology. HR-XML adalah sebuah library yang terdiri dari lebih 75 skema XML yang interindependen, yang memfasilitasi *e-business* dan otomatisasi pertukaran data yang berkaitan dengan sumber daya manusia (*human resources*). HR-XML dikembangkan oleh **HRConsortium**, sebuah organisasi independen dan non profit. Skema XML mendefinisikan elemen data bagi transaksi HR, dan juga memandu penggunaan elemen-elemen tersebut. Berikut adalah bagian dari HR ontology yang digunakan dalam memberikan ilustrasi contoh di atas.

Berikut adalah langkah menghitung besar perubahan yang ada dari keberagaman di atas apabila salah satu dari kedua lokal skema mengalami perubahan:

1. menghitung perbedaan antara dua skema lokal tersebut dengan *common ontology* adalah dengan melihat dan memperhatikan jenis modifikasi, dalam penelitian ini hanya memunculkan 3 (tiga) jenis modifikasi yaitu tambah, hapus dan *rename* kelas dan kelas-properti.



Gambar 3.10: Bagian dari Common Ontology - HR-XML

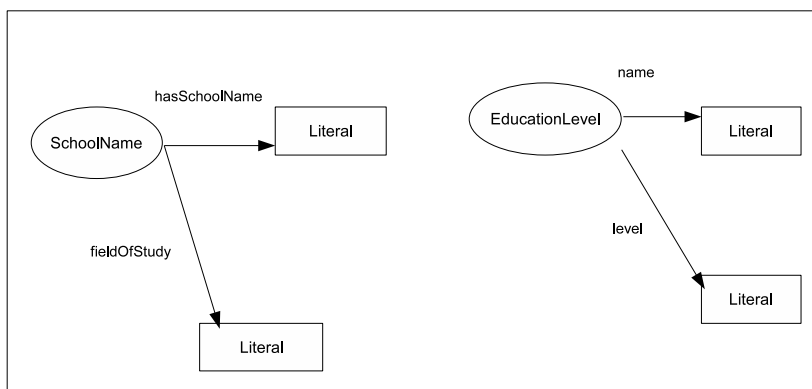
2. Dari hasil langkah pertama akan didapatkan jumlah nilai perbedaannya, kemudian dicari nilai batas untuk dapat memberikan keputusan pada pemilihan jenis algoritma, apakah menggunakan algoritma sederhana (*simple*) atau algoritma kompleks. Nilai batas dicari dan didapat dengan cara *try and error*, dengan melihat hasil *precision*, *recall* dan *f-measure* yang paling optimal.
3. Maka dapat diputuskan algoritma yang digunakan, dengan jumlah nilai perbedaan yang didapat, setelah itu hasil itu dibandingkan dengan pendapat expert. Kemudian dibandingkan antara hasil kalkulasi dengan pendapat expert.

Model kedua yang dapat dilakukan untuk mengetahui algoritma mana yang digunakan adalah dengan membandingkan penggunaan algoritma sederhana (*simple*) dan algoritma kompleks masing masing dengan *expert*. Seperti yang sudah disinggung di muka bahwa algoritma sederhana (*simple*) adalah bagian dari algoritma kompleks, yaitu melakukan perhitungan agreement dengan melakukan *label matching* dan memberikan nilai threshold padanya dan akan dihasilkan sejumlah nilai yang menyatakan apakah suatu kelas / konsep dalam dua skema lokal tersebut similar atau tidak, sehingga dapat dipilih untuk dijadikan sebagai kandidat kelas untuk dilaksanakan proses *mapping*. Apabila

ternyata dalam perhitungan dengan menggunakan algoritma sederhana ditemukan kelas yang ragu ragu, yaitu kelas dari dua skema lokal yang mempunyai nilai similaritas diantara nilai yang dimiliki oleh threshold (threshold-2) untuk mencari hasil dari similaritas dengan nilai threshold untuk memastikan adanya nilai yang masuk kelas kategori ragu ragu (threshold-3), dengan kata lain apabila nilai similaritas lebih besar dari nilai (threshold-2 - threshold-3) dan nilai similaritas kurang dari threshold-2 maka, kelas tersebut termasuk kategori kelas ragu-ragu. Untuk itu harus diperlukan penelusuran mendalam, apakah memang dapat dipilih sebagai kelas kandidat atau tidak.

Untuk hal tersebut akan dilakukan proses perhitungan dengan melanjutkan ke proses yang biasa disebut sebagai *fine tune* dari hasil similaritas yang ada, yaitu melakukan *internal matching* dan *external matching*. Kelas dari skema lokal tersebut yang termasuk kategori ragu ragu, akan dihitung dengan mempertimbangkan atribut atau properti yang dimiliki kelas tersebut, dan properti tersebut dihitung kembali similaritasnya apabila memenuhi kondisi nilai threshold tertentu. Gambar di bawah ini dapat memberikan gambaran contoh dari dua kelas yang dimiliki oleh skema lokal yang termasuk kategori kelas ragu ragu dan kelas sejenis yang terdapat pada common ontology.

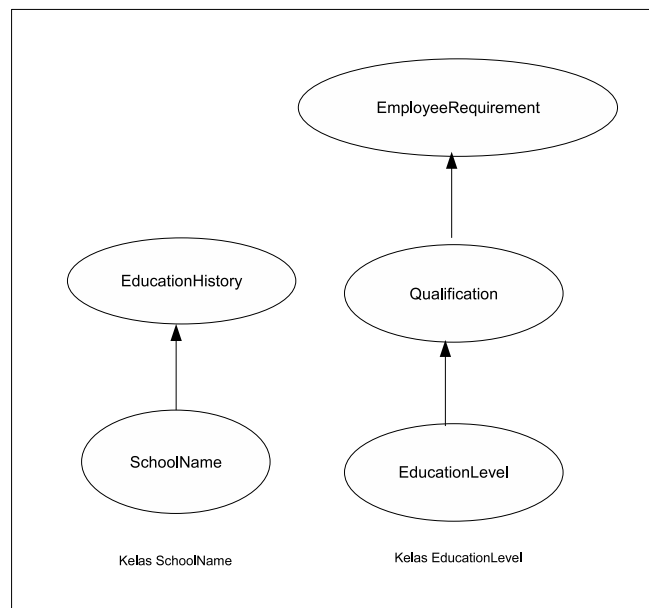
---



Gambar 3.11: Kelas Ragu Ragu

Setelah mendapatkan nilai pada saat melakukan *internal matching*, maka dilanjutkan dengan menghitung menggunakan *external matching*, yaitu dengan mempertimbangkan keberadaan kelas tersebut dalam hirarkinya, yaitu melihat *upper-class* yang dimiliki oleh masing masing kelas. Gambar berikut dapat menunjukkan masing-masing *upper-class* dari *common ontology* dan skema lokal dari kelas kategori ragu ragu.

---



Gambar 3.12: UpperClass dari Kelas Ragu Ragu

## Bab IV

# Hasil Pengujian Pendekatan

### 4.1 Pendahuluan

Dalam bab ini, akan digambarkan implementasi dari uji coba pertimbangan pemeliharaan pendekatan *semantic agreement*. Mengacu kepada proses umum *peer agreement* yang telah dilakukan dalam pendekatan yang dikembangkan Wicaksana [Wicaksana, 2006].

### 4.2 Persiapan Pengujian Sederhana

Pada bagian ini, akan dijelaskan mengenai detail dari persiapan pengujian. Persiapan pertama, pembuatan common ontology dilakukan oleh sekumpulan orang yang ahli dalam bidang *Human Resource Development*(HRD) khususnya di layanan job matching, walaupun, seperti sudah dikemukakan di bab 1 bahwa sebagai common ontology diambil dari HR-XML.org, tetapi untuk uji coba hanya diacu sebagian ontologi HR-XML yang sesuai dengan kebutuhan. Ahli di sini hanya untuk konfirmasi mengenai kevalidan *common ontology* yang telah dibuat untuk pengujian ini. Selanjutnya adalah menyiapkan skema lokal sejumlah 10. Skema lokal tersebut menggambarkan *peer* yang terlibat dalam *job matching*.

### 4.3 Pelaksanaan Pengujian Sederhana

Beberapa skenario pengujian dilakukan dalam rangka menguji model usulan pertimbangan pemeliharaan pendekatan *semantic agreement*, yaitu melakukan perubahan pa-

da salah satu skema atau lebih , secara detil skenario perubahan adalah sebagai berikut:

1. *Local Schema* dari peer akan diubah pada dua kelas, dua kelas dan dua properti dan seterusnya dibuat jumlah kelas dan properti yang lebih banyak.
2. *Local Schema* dari peer akan ditambah dengan menambahkan dua kelas, dua kelas dan dua properti dan seterusnya dibuat jumlah kelas dan properti yang lebih banyak.
3. *Local Schema* dari peer akan dihapus dengan menghapus dua kelas, dua kelas dan dua properti dan seterusnya dibuat jumlah kelas dan properti yang lebih banyak.

Untuk mengevaluasi beberapa skenario yang berbeda di atas, harus melihat hasilnya sehingga dapat dilakukan pertimbangan untuk perhitungan *agreement* apakah akan dilakukan dengan algoritma yang kompleks atau algoritma sederhana. Kemudian hasil yang ada akan dibandingkan dengan ahli di bidang tenaga kerja dan pendidikan. Alat ukur yang digunakan dalam mengevaluasi hasil dari pengujian adalah *Recall*, *Precision* dan *F-measure* [Ioannis, 2005]. Berikut adalah definisi dan persamaannya:

*Recall* adalah proporsisi dari dokumen yang relevan, di luar dari seluruh dokumen yang relevan yang tersedia.

$$Recall = \frac{|\{relevantdata\} \cap \{retrieveddata\}|}{|\{relevantdata\}|} \quad (IV.1)$$

*Precision* adalah proporsi dari dokumen yang di-retrieve dan relevan untuk seluruh dokumen yang di-retrieve.

$$Precision = \frac{|\{relevantdata\} \cap \{retrieveddata\}|}{|\{retrieveddata\}|} \quad (IV.2)$$

*F-measure* adalah rata-rata harmoni bobot dari precision dan recall.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (IV.3)$$

Berdasarkan skenario pengujian di atas, dapat disimpulkan bahwa terdapat enam model pengujian. Berikut enam model pengujian tersebut serta hasil perhitungannya, dengan bantuan tool Protege, PromptTab (implementasi dari algoritma PromptDiff) plug-in yang terdapat pada Protege, aplikasi online *semantic similarity* berbasis WordNet, bahasa pemrograman Python dan MS.Excell

- Menambah kelas ,
- Menambah Kelas dan Properti,
- Menghapus kelas,
- Menghapus kelas dan properti,
- Mengubah kelas,
- Mengubah kelas dan properti.

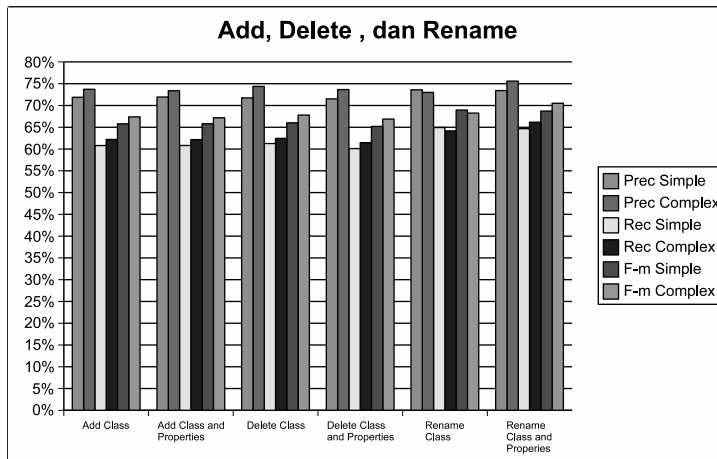
## 4.4 Hasil dan Diskusi

Hasil dari perhitungan perbedaan untuk ke sepuluh (10) lokal skema menghasilkan informasi bahwa hanya operasi *rename* pada kelas, yang akan dapat menggunakan algoritma sederhana untuk proses pemeliharaan *agreement* nya. Untuk operasi modifikasi lainnya, kesemuanya, usulan yang dihasilkan adalah menggunakan algoritma kompleks.

Hasil dari perhitungan dengan menggunakan model ke dua, yaitu membandingkan antara algoritma sederhana dan kompleks, dengan memperhatikan pendapat dari expert, adalah sebagai berikut:

- rangkuman eksperimen dari semua skema lokal (10) untuk semua skenario ( 5 skenario) yang sudah dilaksanakan dapat dilihat pada gambar 4.1. Pada hasil tersebut terlihat modifikasi *rename* pada kelas menghasilkan nilai *f-measure* yang tertinggi. Hal itu menandakan bahwa pengubahan nama kelas tidak mempengaruhi *agreement* yang dihasilkan karena tidak merubah struktur, sehingga dapat diusulkan hanya menggunakan algoritma sederhana saja, selain operasi itu, operasi modifikasi lainya diusulkan menggunakan algoritma kompleks.
- rangkuman hasil eksperimen dari semua skema lokal (10) untuk berbagai modifikasi, lihat gambar 4.2. Dapat dikatakan bahwa ternyata skema lokal ke 4 (LS 4) mempunyai nilai *F-measure* tertinggi. Hal tersebut menandakan bahwa skema lokal LS4 mempunyai konsep yang sebagian besar sesuai dengan *common ontology* yang tersedia walaupun terjadi beberapa modifikasi padanya.
- rangkuman hasil eksperimen dari semua skema lokal (10) untuk berbagai modifikasi dapat dilihat pada gambar 4.3. Pada grafik tersebut terlihat bahwa skenario

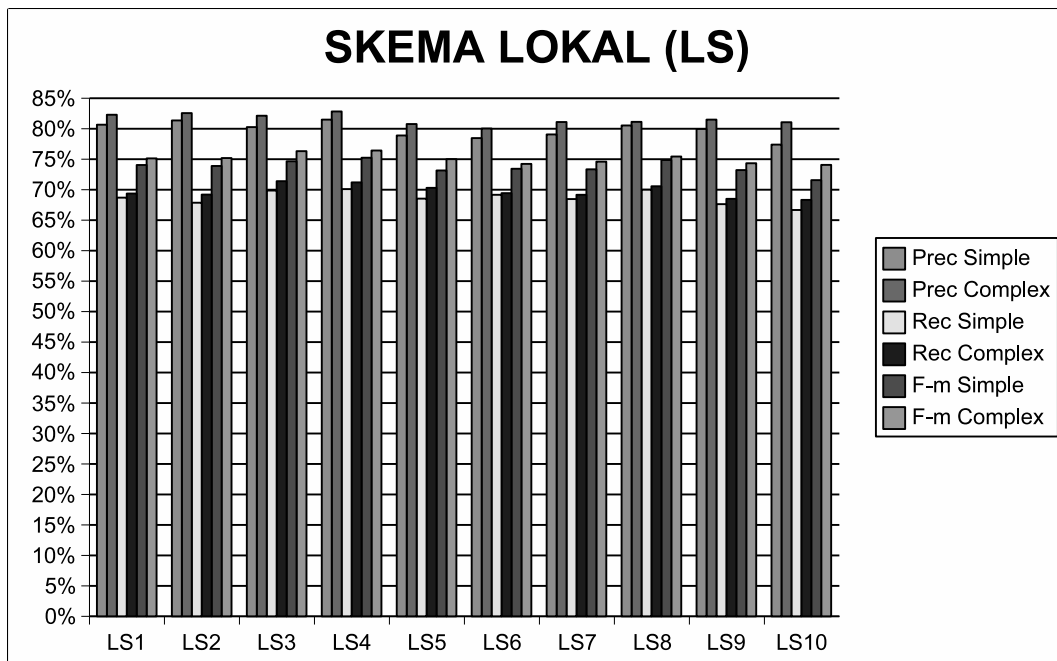
pertama (1) yaitu penambahan, penghapusan dan *rename* untuk dua (2) kelas dan 2 kelas, 2 properti memberikan hasil *f-measure* tertinggi, yang berarti bahwa dengan makin sedikit perubahan atau modifikasi yang dilakukan, maka akan semakin mudah untuk dapat dipelihara dan menyesuaikan dengan *common ontology* nya.



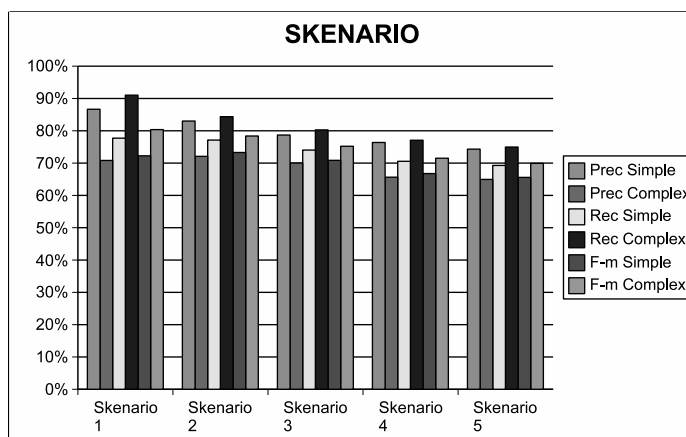
Gambar 4.1: Modifikasi *Add*, *Delete* dan *Rename* pada 10 LS

Berdasarkan hasil uji coba yang telah dilakukan, terdapat beberapa hal yang menjadi catatan, yaitu:

- F-measure pada operasi modifikasi untuk skema lokal menunjukkan bahwa perhitungan untuk pemeliharaan agreement pada modifikasi rename memberikan nilai paling tinggi dibandingkan operasi add dan delete, yaitu 0.69 yang berarti menunjukkan kedekatan perhitungan dengan pendapat expert, pada domain layanan job matching.
- F-measure pada setiap skema lokal menunjukkan bahwa perhitungan untuk pemeliharaan agreement pada skema lokal LS4 memberikan nilai paling tinggi, yaitu 0.75.



Gambar 4.2: Skema Lokal untuk Masing-masing Skenario



Gambar 4.3: Rangkuman Seluruh Skenario Modifikasi untuk 10 LS

- F-measure pada setiap skenario yang ada (5 skenario) menunjukkan bahwa perhitungan untuk pemeliharaan agreement pada skenario 1 (satu) memberikan nilai

paling tinggi, yaitu 0.78.

- Hasil pendekatan ini dapat memberikan kontribusi untuk pengembangan pendekatan pemeliharaan *agreement*.

## Bab V

# Kesimpulan dan Rencana Ke Depan

### 5.1 Umum

Hal pertama adalah terdapatnya kesimpulan yang mengacu kepada tujuan dari penelitian dalam pertimbangan pemeliharaan agreement pada lingkungan P2P yang diterapkan pada layanan job matching. Yang ke dua, ringkasan dari pencapaian usulan pertimbangan pemeliharaan agreement. Yang terakhir, adanya kontribusi utama dari disertasi ini dan mendiskusikan langkah apa yang sebaiknya dilakukan pada penelitian berikutnya.

### 5.2 Tujuan Penelitian

Tujuan utama dari penelitian ini adalah mengembangkan satu pendekatan untuk mengetahui perubahan skema lokal atau ontologi sehingga perlu melakukan pemeliharaan agreement antaran suatu common ontology dan provider peer yang ada. Dalam lingkungan yang dinamis dan otonom, administrator harus secara terus menerus memonitor common ontology. Monitoring secara manual adalah sangat mahal dan tidak akan dapat ter-skala dengan baik apabila jumlah peer meningkat. Hal yang penting adalah mengembangkan teknik yang secara otomatis memonitor dan melakukan usulan perbaikan agreement.

## 5.3 Hasil

Ringkasan dari Hasil dan diskusi. Hasil pendekatan ini dapat memberikan kontribusi untuk pengembangan pendekatan pemeliharaan agreement. Ternyata tidak semua jenis modifikasi terhadap skema lokal yang ada, tidak harus menggunakan algoritma kompleks untuk proses pemeliharaan, tetapi untuk jenis perubahan tertentu yaitu *rename* pada kelas algoritma sederhana dapat digunakan untuk pemeliharaan agreement.

## 5.4 Kontribusi

Hasil utama dari penelitian ini adalah spesifikasi dari suatu framework berbasis pada satu pendekatan semantic agreement untuk lingkungan P2P. Framework ini didasarkan atas arsitektur P2P model hybrid dua level yang telah dikembangkan oleh [Wicaksana, 2006] yang terdiri dari dua tipe peer: (1) super peer yang digunakan untuk me-register dan mengelola peer lainnya, dan (2) simple peer yang isinya diekspor dan dibagi dengan peer yang lain. Penelitian ini mengembangkan suatu model untuk mengusulkan pemeliharaan semantic agreement pada lingkungan P2P, sehingga pendekatan sebelumnya yang cenderung statis akan menjadi dinamis karena pendekatan ini akan dapat mendeteksi perubahan yang terjadi dalam agreement dan melakukan usulan penyesuaian terhadapnya.

Telah dikembangkan prototipe simulasi untuk mengevaluasi dan memvalidasi beberapa konsep dan ide yang didiskusikan dalam disertasi ini. Sebagai ringkasan, kontribusi utama dari disertasi ini adalah : mengembangkan satu model usulan pertimbangan apakah pendekatan mediasi yang menggunakan *semantic agreement* yang telah dikembangkan Wicaksana dapat terpelihara, Sehingga didapatkan usulan pertimbangan apakah akan menggunakan pendekatan *semantic agreement* yang kompleks atau yang sederhana.

## 5.5 Rencana Ke Depan

Dari kegiatan penelitian yang telah dilakukan, terdapat kesempatan untuk mengembangkan lebih lanjut, yaitu:

- Memperluas domain pengujian.

- Mengembangkan algoritma sederhana sebagai alternatif algoritma kompleks karena hasil perbandingan perubahan skema atau ontologi untuk pemeliharaan agreement

# Daftar Pustaka

- [Arumugam et al., ] Arumugam, M., Sheth, A., and Arpinar, B. I. The Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web.
- [Aspnes and Shah, ] Aspnes, J. and Shah, G. Skip graphs. *CoRR*, cs.DS/0306043.
- [Avaki, 2001] Avaki (2001). <http://www.avaki.com/>.
- [Bacastow and Lewis, ] Bacastow, T. S. and Lewis, B. Peer-to-Peer GIS. <http://www.personal.psu.edu/faculty/t/s/tsb4/docs/p2pgis.doc>.
- [Banowosari and Wicaksana, 2007] Banowosari, L. Y. and Wicaksana, I. W. S. (2007). Evaluation two label matching approaches for indonesian language. In *Proceeding SNIKTI VII UI 2007*. Universitas Indonesia, Universitas Indonesia.
- [Berners-Lee and Connolly, 1998] Berners-Lee, T. and Connolly, D. (1998). Web Architecture: Extensible Language.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lasila, O. (2001). The semantic web. *Scientific American*.
- [Bishr, 1997] Bishr, Y. (1997). *Semantic Aspects of Interoperable GIS*. PhD thesis, Wageningen Agricultural University, Netherland.
- [Bizer et al., 2005] Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., and Eckstein, R. (2005). *The Impact of Semantic Web Technologies on Job Recruitment Process*, chapter 15, pages 1367–1381. Physica-Verlag HD. Springer Link.
- [Borden, 2007] Borden, J. (2007). <http://www.stylusstudio.com/xmldev/200310/post10060.html>.
- [Bourse et al., 2002] Bourse, M., Harzallah, M., Lecler, M., and Trichet, F. (2002). COMMONCV: modeling the competencies underlying a Curriculum Vitae. In *In Proc. ACM International Conf Proc Series*, volume 27.

- [Budanitsky and Hirst, ] Budanitsky, A. and Hirst, G. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.
- [Cali et al., ] Cali, A., Calvanese, D., Giacomo, G. D., and Lenzerini, M. on the role of integrity constraints in data integration. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.
- [Crespo and Garcia-Molina, ] Crespo, A. and Garcia-Molina, H. Routing indices for peer-to-peer systems. In *ICDCS*, pages 23–.
- [Entropia, 2004] Entropia (2004). <http://www.entropia.com/>.
- [Euzenat et al., 2004] Euzenat, J., Le Bach, T., Barasa, J., and etc (2004). D2.2.3: State of the art on ontology alignment. Technical Report IST-2004-507482, knowledgeweb.
- [Euzenat and Valtech, ] Euzenat, J. and Valtech, P. Similarity-Based Ontology Alignment in OWL-Lite. In *ECAI*, pages 333–337.
- [Freedman and Mazières, ] Freedman, M. J. and Mazières, D. Sloppy hashing and self-organizing clusters. In *IPTPS*, pages 45–55.
- [Freenet, 1999] Freenet (1999). <http://freenetproject.org/>.
- [Giunchiglia and Zaihrayeu, ] Giunchiglia, F. and Zaihrayeu, I. Implementing database coordination in P2P networks. Technical report, University of Trento.
- [Gnutella, 2003] Gnutella (2003). <http://www.gnutella.com>.
- [Gruber, 1995] Gruber, T. (1995). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *Int. Journal of Human-Computer Studies*, 43:907–928.
- [Gupta et al., ] Gupta, I., Birman, K. P., Linga, P., Demers, A. J., and van Renesse, R. Kelips: Building an efficient and stable p2p dht through increased memory and background overhead. In *IPTPS*, pages 160–169.
- [Haase and Stojanovic, 2005] Haase, P. and Stojanovic, L. (2005). Consistent evolution of OWL ontologies. In *In Proc. of the Second European Semantic Web Conference (ESWC'05)*, volume 3532 of *Lecture Notes in Computer Science*, pages 182–197.
- [Harvey et al., ] Harvey, N. J. A., Jones, M. B., Saroiu, S., Theimer, M., and Wolman, A. Skipnet: A scalable overlay network with practical locality properties. In *USENIX Symposium on Internet Technologies and Systems*.

- [Heflin et al., 2003] Heflin, J., Hendler, J., and Luke, S. (2003). SHOE: a Blueprint for the Semantic Web. [www.cse.lehigh.edu/heflin/pubs/swbook03.pdf](http://www.cse.lehigh.edu/heflin/pubs/swbook03.pdf).
- [Huang and Stuckenschmid, 2005] Huang, Z. and Stuckenschmid, H. (2005). Reasoning with multi-version ontologies: a temporal logic. In *In Proc. of the 4th International Semantic Web Conference (ISWC 2005)*, pages 398–412, Galway, Ireland.
- [Ioannis, 2005] Ioannis, V. (2005). Semantic Similarity Methods in WOrdNet and Their Application to Information Retrieval on The Web. In *Proc of the 7th ACM International workshop on Web Information and Data Management*, pages 1–10, Bremant, Germany.
- [Jawhar and Wu, 2004] Jawhar, I. and Wu, J. (2004). A Two-Level Random Walk Search Protocol for Peer-to-Peer Networks. In *Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics*.
- [JXTA, 2004] JXTA (2004). <http://www.jxta.org/>.
- [Kalfoglou and Schorlemmer, ] Kalfoglou, Y. and Schorlemmer, M. Ontology Mapping: The State of the Art. In *Semantic Interoperability and Integration*.
- [Kalogeraki et al., ] Kalogeraki, V., Gunopulos, D., and Zeinalipour-Yazti, D. A local search mechanism for peer-to-peer networks. In *CIKM*, pages 300–307.
- [Klein and Bernstein, 2003] Klein, M. and Bernstein, A. (2003). Searching for Services on the Semantic Web Using Process Ontologies.
- [Klein et al., 2002] Klein, M., Fensel, D., Kinyakof, A., and Ognyanoy, D. (2002). Ontology versioning and change detection on the web. In *In Proc. of the 13th conf on Knowledge Engineering and Knowledge Management - Ontologies and the Semantic Web (EKAW 2002)*, volume 2437 of *Lecture Notes in Computer Science*, pages 197–212. Springer.
- [Klein and Noy, 2003] Klein, M. and Noy, N. (2003). A Component Based Framework for Ontology Evolution. In *In Proc. of IJCAI 2003 Workshop on Ontologies and Distributed Systems*.
- [Lv et al., 2001] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. (2001). Search and replication in unstructured peer-to-peer network.

- [Mackay, 1999] Mackay, D. S. (1999). Semantic Integration of Environmental Models for Appliatio to Global Information Ssytems and Decision Making. *SIGMOD Record*, 28(1).
- [Madhavan et al., ] Madhavan, J., Bernstein, P. A., and Rahm, E. Generic Schema Matching with CUPID. In *Proc. of 27th Intl. Conf. on Very Large Data Bases, Very Large Data Bases*, pages 49–58.
- [Maedche et al., ] Maedche, A., Motik, B., Silva, N., and Volz, R. MAFRA - A MAPping FRamework for Distributed Ontologies. In *Proc. of 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW)*.
- [Manber, 1999] Manber, U. (1999). *Modern Information Retrieval*. Addiso Wesley.
- [Manku et al., 2003] Manku, G., Bawa, M., and Raghavan, P. (2003). Symphony: Distributed Hashing in a Small World.
- [Milojicic et al., 2002] Milojicic, D., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollinx, S., and Xu, Z. (2002). Peer-to-peer Computing. [www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf](http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf).
- [Napster, 2003] Napster (2003). <http://www.napster.com/>.
- [Noy, 2003] Noy, N. F. (2003). Semantic Integration: a Survey of Ontology-Based Approaches. In *Proc. of Workshop on Semantic Integration at SWC-2003*, number 13 in Semantic Integration, Sanibel Island, FL.
- [Noy and Musen, ] Noy, N. F. and Musen, M. A. Smart: Automated Support for Ontology Merging and Alignment. In *Proc. of the 12th Workshop on Knowledge Acquisition, Modelling, and Management (KAW'99)*, Banf, Canada.
- [Paoli and Disco, 1005] Paoli, F. D. and Disco, M. L. (1005). Enhancing Semantic Interoperability in Collaborative Systems. University of Milano Bicocca, Italy.
- [Pitoura et al., 2003] Pitoura, E., Abiteboul, S., Pfoser, D., Samaras, G., and Vazirgiannis, M. (2003). DBGlobe: A Service-Oriented P2P System for Globa Computing. *Sigmond Record*, 32(3):77–82.
- [Plessers et al., 2007] Plessers, P., Troyer, O. D., and Casteleyn, S. (2007). Understanding Ontology Evolution: A Change Detecetion Approach. *Journal of Web Semantic*, 5:39–49.

- [Rahm and Bernstein, ] Rahm, E. and Bernstein, P. A. A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 10:334–250.
- [Rhea and Kubiawicz, ] Rhea, S. C. and Kubiawicz, J. Probabilistic location and routing. In *INFOCOM*.
- [Rowstron and Druschel, ] Rowstron, A. and Druschel, P. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed System Platforms (Middleware 2001)*, Heidelberg, Germany.
- [SETI@Home, 2001] SETI@Home (2001). <http://setiathome.ssl.berkeley.edu/>.
- [Silaghi et al., ] Silaghi, B. D., Gopalakrishnan, V., Bhattacharjee, B., and Keleher, P. J. Hierarchical routing with soft-state replicas in terradir. In *IPDPS*.
- [Sinir, ] Sinir, S. S. Ontology Mapping Survey (Slide).
- [Staab, 2006] Staab, S. (2006). *Semantic Web and Peer-to-Peer*. Springer Verlaag.
- [Stoica et al., 2001] Stoica, I., Morris, R., Karger, D., Kaashoek, M., and Balakrishnan, H. (2001). Chord: A Scalable Peer-to-peer Lookup Service for Internet Applciation. In *Proc. of SIGCOMM'01*, San Diego, California, USA.
- [Stojanovic, 2004] Stojanovic, L. (2004). *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, Germany.
- [Stojanovic et al., 2002] Stojanovic, L., Maedche, A., Motik, B., and Stojanovic, N. (2002). User driven ontology evolution management. In *In Proc. of the Thirteenth European Conference on Knowledge Engineering and Knowledge management EKAW*, pages 285–300, Madrid, Spain.
- [Tanin et al., ] Tanin, E., Brabec, F., and Samet, H. Remote access to large spatial databases. *GIS*, pages 5–10.
- [Tempich et al., ] Tempich, C., Staab, S., and Wranik, A. REMINDIN: semantic query routing in Peer-to-Peer networks based on social metaphors. In *Proceeding of WWW2004*, New York, USA.
- [Tsoumakos and Roussopoulos, ] Tsoumakos, D. and Roussopoulos, N. Adaptive probabilistic search for peer-to-peer networks. In *Peer-to-Peer Computing*, pages 102–109.

- [Visser and Schuster, ] Visser, U. and Schuster, G. Finding and Integration of Information a Proactical Solution for the Semantic Web. In Euzenat, J., Gomez-Perez, A., Guarino, N., and Stuckenschmidt, H., editors, *Proceedings of ECAI 02, Workshop on Ontologies and Semantic Interoperability*, pages 73–78, Lyon, France.
- [Vogele and Schlieder, ] Vogele, T. and Schlieder, C. The Use of Spatial Metadata for Information Retrieval in Peer-to-Peer Networks. In *Proc. of 5th AGILE Conference Geographic Informatin Science*, Palma, Balearic Island, Spain.
- [Vrije Universiteit, 2004] Vrije Universiteit (2004). Ontology Management (Del4): Survey, Requirement and Direction. web of the university.
- [Warin, 2004] Warin, M. (2004). Using WordNet and Semantic Similarity to Disambiguate an Ontology. internet.
- [Wicaksana, 2006] Wicaksana, I. W. S. (2006). *Peer-to-peer Information Interoperability: A Semantic Ageement Based Mediation Approache*. PhD thesis, Universite de Bourgogne, Dijon, France.
- [Wicaksana and Guritno, 2005] Wicaksana, I. W. S. and Guritno, S. (2005). Berbagi Informasi Tenaga Kerja dalam e-Government dengan Infrastruktur P2P. In *In Porc. Seminar Nasional ITS*, Surabaya, Indonesia.
- [Yang and Garcia-Molina, ] Yang, B. and Garcia-Molina, H. Designing a super-peer network. (2002-13).
- [Yang and Wu, 2003] Yang, C. and Wu, J. (2003). A Dominating-set-based Routing in Peer-to-Peer Networks. In *Proc. of the 2nd International Workshop on Grid and Cooperative Computing Workshop (GCC'03)*.
- [Yildiz, 2006] Yildiz, B. (2006). Ontology Evolution and Versionng The state fo the art. Vienna University of Technology, Institute of Softwrae Techology & Interactive Systems (ISIS), [http://publik.tuwien.ac.at/files/pub-inf\\_4603.pdf](http://publik.tuwien.ac.at/files/pub-inf_4603.pdf).
- [Zhang et al., ] Zhang, H., Goel, A., and Govindan, R. Using the small-world model to improve freenet performance. *Computer Networks*, 46(4):555–574.