

## JARIS : PROTOKOL QoS PERPUSTAKAAN DIGITAL BERBASIS PEER-TO-PEER

Heri Kurniawan, dan Zainal A. Hasibuan

Fakultas Ilmu Komputer

Universitas Indonesia

E-mail: herik@ui.edu, zhasibua@cs.ui.edu

### ABSTRAKSI

Pencarian dokumen pada jaringan Perpustakaan Digital (PD) berbasis peer-to-peer dengan arsitektur terdistribusi sangat mengandalkan koneksi antar peer. Salah satu teknik pencarian yang umum digunakan adalah melalui pengiriman kueri secara broadcast kepada peer tetangga. Sayangnya cara ini kurang efisien karena berpotensi meningkatkan kepadatan lalu lintas jaringan. Selain permasalahan efisiensi, permasalahan lain yang dihadapi aplikasi PD adalah rendahnya kredibilitas PD dalam pertukaran data. Hal ini dapat dilihat dari rendahnya kualitas servis yang diberikan oleh sebuah PD. Pada penelitian ini penulis mengusulkan sebuah metode yang bernama jaris. Jaris melakukan pencarian berdasarkan kemiripan antar PD pada cache cluster dan kemiripan kueri pada cache kueri. Cara ini dilakukan untuk meningkatkan relevansi hasil pencarian dokumen dan menurunkan jumlah pesan yang dikirim. Jaris menggunakan mekanisme polling yang menggabungkan nilai QoS lokal dan global untuk meningkatkan kualitas transaksi. Pada uji coba penelitian, metode jaris dibandingkan dengan metode random voting. Hasil percobaan menunjukkan kinerja jaris lebih baik bila dibandingkan dengan random voting.

**Kata Kunci:** P2P, Perpustakaan Digital.

### 1. PENDAHULUAN

Teknologi Peer-to-peer (P2P) merupakan media pertukaran data yang populer dikalangan pengguna jaringan. P2P memanfaatkan kekuatan komputasi dan *bandwidth* jaringan pengguna dalam aktifitas pertukaran file [11]. Salah satu aplikasi pertama yang menggunakan teknologi P2P adalah Napster [10]. Napster banyak disukai pada akhir tahun 90an. Saat itu pertukaran file musik berlangsung dengan mudah secara *point-to-point*. Kemudahan ini membuat publik tertarik dan kemudian bergabung untuk melakukan pertukaran data.

Saat ini aplikasi P2P dapat juga diaplikasikan pada perpustakaan digital (PD). Beberapa aplikasi PD tersebut antara lain Freelib [2], P2PDL [16], Bricks [12] dan P2P4DL [15]. Aplikasi-aplikasi PD ini dijalankan pada sejumlah terminal atau komputer yang tergabung dalam jaringan P2P. Dalam konteks P2P, terminologi terminal disebut juga dengan peer [11]. Pada konteks penelitian ini, terminologi peer penulis sebut sebagai PD.

Setiap PD mempunyai profil atau karakter yang berbeda dalam jaringan P2P. Penulis mengasumsikan empat faktor yang mempengaruhi kualitas servis sebuah PD. Keempat faktor tersebut adalah :

1. Kecepatan akses, besaran *bandwidth* yang dimiliki oleh sebuah PD.
2. Kualitas dokumen, dokumen tidak mengandung virus atau berkualitas rendah.
3. Jumlah koleksi, banyaknya koleksi yang dimiliki oleh PD.

4. Jumlah interaksi, jumlah interaksi PD lain dengan PD tersebut, atau bisa disebut juga popularitas PD diantara PD lain.

Keempat faktor diatas, penulis asumsikan sebagai penentu kualitas layanan PD. Pada penelitian ini, kualitas layanan PD direpresenstasikan dalam bentuk nilai QoS. Setiap PD mempunyai nilai QoS yang berbeda. Nilai QoS diperoleh setelah sebuah PD melakukan pertukaran data dengan PD lain. Rekaman nilai yang disimpan menjadi acuan bagi sebuah PD bila ingin melakukan pertukaran data selanjutnya.

Pada kenyataannya tidak semua PD mempunyai nilai QoS yang baik. Sifat lingkungan P2P yang bebas tanpa pengawasan terpusat, tidak lepas dari aktifitas sejumlah *malicious PD* yang mencoba melakukan kegiatan yang dapat mengganggu PD lain. Aktifitas umum yang dilakukan oleh *malicious PD* adalah pertama memberikan dokumen yang dapat mengganggu sistem pengguna. Dokumen ini dapat mengandung virus, trojan, ataupun worm yang dapat merusak sistem pengguna [9]. Aktifitas kedua adalah melakukan *colluding* [5], [3], [6], yaitu memberikan respon nilai QoS rendah bagi PD yang sebenarnya mempunyai nilai QoS tinggi dan memberikan nilai QoS tinggi bagi sesama *malicious PD*. Umumnya *colluding* dilakukan saat polling dilakukan. Polling merupakan salah satu teknik pengambilan informasi nilai QoS yang dimiliki sebuah PD dengan cara menanyakan pengalaman PD lain tentang dengan PD yang akan diakses. Walau jaringan dipenuhi

dengan aktifitas *malicious PD*, mayoritas pengguna PD menginginkan kerjasama dengan PD yang benar-benar mempunyai QoS baik. Sayangnya hingga saat ini belum ada mekanisme khusus yang mengatur QoS PD dalam jaringan P2P. Padahal mekanisme ini dapat menjadi bahan pertimbangan bagi sebuah PD sebelum mengambil data dari PD lain.

Pada penelitian ini, penulis mencoba mengusulkan manajemen QoS PD berbasis P2P yang bernama Jaris. Secara umum, skenario dalam metode Jaris terdiri dari tiga tahap yaitu inialisasi kemiripan antar peer, pencarian dokumen dan *polling*.

## 2. LANDASAN TEORI

### 2.1 Teknologi Peer-to-Peer

Peer-to-peer (P2P) merupakan salah satu teknologi komputasi terdistribusi yang memberi kemudahan sebuah peer untuk melakukan koneksi dengan peer lain secara langsung [8]. Istilah peer merepresentasikan perangkat (komputer, PDA, dan lain) yang terhubung dalam jaringan P2P. Umumnya P2P digunakan sebagai protokol pencarian dan pertukaran data yang tersebar diberbagai lokasi. Berbeda dengan mekanisme client-server, dalam P2P tidak ada peer yang bertindak selayaknya server yang memonitor atau mengontrol peer lain dalam jaringan. Setiap peer mempunyai kebebasan untuk masuk dan keluar dari jaringan tanpa harus memperoleh izin dari peer lain. Kemudahan ini yang membuat P2P makin digemari dan digunakan oleh banyak kalangan.

Secara umum tujuan penggunaan aplikasi P2P dapat dibagi menjadi tiga klasifikasi [6]:

#### 1. Pertukaran data

Aplikasi P2P seperti Napster, Morpheus, Freenet, KaZaA dan yang lain memberi fasilitas untuk pertukaran data.

#### 2. Proses terdistribusi

Teknologi P2P dapat digunakan dalam pendistribusian proses komputasi kepada peer lain seperti pada proyek SETI@Home.

#### 3. Instant Messaging

Program *chatting* seperti MSN Messenger, Yahoo! Messenger menggunakan teknologi P2P sehingga user dapat saling bertukar pesan dan file.

Dari sisi arsitektur, jaringan P2P dibagi menjadi tiga, yaitu terpusat (napster), tersebar (Gnutella) dan hybrid (gabungan terpusat dan tersebar). Kategori pertama menggunakan server terpusat untuk menyimpan file indeks dari informasi file yang dimiliki setiap peer dalam jaringan. Pada kategori kedua, komunikasi antar peer dan pertukaran file terjadi tanpa campur tangan server

pusat. Kemudian pada kategori ketiga adalah sejumlah peer dibagi menjadi *leaf* dan *superpeer*.

### 2.2 Perpustakaan Digital

Perpustakaan digital mempermudah peneliti maupun profesional dalam memenuhi kebutuhan informasi tanpa batasan tempat dan waktu. Saat ini layanan perpustakaan digital umumnya menggunakan model *client-server*. Model *client-server* memberikan keuntungan dari sisi pengaturan pembaharuan dokumen dan kebijakan keamanan, namun disisi lain model ini membuat beban kerja server bertambah dan rendahnya efisiensi pemanfaatan sumber daya jaringan (seperti kapasitas penyimpanan dan *bandwith*). Teknologi P2P hadir memberikan sejumlah kemampuan yang berbeda dengan arsitektur *client-server*. Melalui teknologi ini diharapkan pemanfaatan sumberdaya jaringan menjadi lebih tinggi dan pengguna menjadi lebih leluasa dalam mengontrol objek digital yang mereka publikasikan. Saat ini beberapa kalangan telah membuat aplikasi perpustakaan digital berbasis peer-to-peer.

Salah satu aktifitas yang sering dilakukan dalam perpustakaan digital adalah pencarian dokumen. Pencarian dokumen dalam PD berbasis P2P terjadi bukan hanya pada satu *host*, namun pada semua *host* yang tergabung dalam komunitas. Penentuan relevansi antara kueri dan dokumen dapat dilakukan dengan menggunakan teknik dalam bidang *information retrieval*. Salah satunya yaitu model *vector space*, untuk mengukur kemiripan antara kueri dengan dokumen.

Model vektor merupakan metode pencarian dokumen dengan merepresentasikan kata-kata dalam dokumen dan kueri melalui bentuk vektor [13]. Pemodelan vektor terdiri dari tiga tahapan, pertama adalah melakukan pengindeksan, yaitu dengan mengekstraksi kata dari dokumen teks. Tahap kedua adalah pemberian bobot terhadap kata dalam indeks untuk meningkatkan relevansi pencarian. Terakhir adalah melakukan pengurutan dokumen berdasarkan hasil pengukuran kemiripan (*similarity measure*) antar kueri dan dokumen.

Model *vector space* digunakan untuk menghitung kemiripan dokumen yang terdapat pada sejumlah PD. Vektor yang merepresentasikan ciri dokumen pada sejumlah *peer* diukur satu sama lain dengan menggunakan *cosine similarity*. Jika nilai kemiripan dokumen antar dua peer besar, maka dapat dikatakan kedua peer tersebut mempunyai dokumen yang saling berkaitan.

### 2.4 Kemiripan Antar Peer

Beberapa penelitian mengenai kemiripan antar *peer* pernah dilakukan, diantaranya oleh Sripanidkulchai [14]. Menurut Sripanidkulchai, jika seorang pengguna tertarik tentang suatu informasi

yang berada pada sebuah peer, maka kemungkinan besar pengguna tersebut juga tertarik pada informasi lain yang berada pada peer tersebut. Sripanidkulchai melakukan eksperimen dengan membuat daftar *peer* yang saling mempunyai kemiripan dalam jaringan Gnutella. Berdasarkan hasil penelitian, ia mengungkapkan bahwa pembuatan daftar *peer* yang saling berkorelasi terbukti mampu meningkatkan peningkatan kinerja pencarian pada jaringan Gnutella.

Pembuatan daftar kemiripan antar *peer* yang telah dilakukan Sripanidkulchai dapat diimplementasikan dengan mengukur perbandingan jarak kata pada masing-masing dokumen yang dimiliki *peer*. Jarak antar kata dapat dimodelkan melalui metode model *vector space*. Melalui model ini, jarak kata yang dihasilkan dapat menjadi input bagi pengelompokan dokumen menjadi  $k$  cluster. Dokumen-dokumen dalam satu cluster mempunyai kemiripan yang lebih tinggi satu sama lain bila dibandingkan dengan dokumen pada cluster lain.

## 2.6 Manajemen Reputasi

Ide manajemen QoS pada penelitian ini berasal dari manajemen reputasi. Manajemen reputasi merupakan proses perekaman transaksi yang dilakukan suatu pihak dan opini pihak lain tentang transaksi tersebut. Hasil rekaman ini berperan sebagai bahan pertimbangan bagi pihak lain untuk memutuskan apakah pihak tersebut dapat dipercaya atau tidak. Dalam P2P, beberapa penelitian mengenai manajemen reputasi telah dikembangkan untuk menjaga dan meningkatkan kepercayaan antar *peer* sebelum melakukan pertukaran data. Salah satu contoh situs yang mengimplementasikan manajemen reputasi adalah eBay [7]. Dalam eBay, penjual dan pembeli saling memberikan rating pada akhir transaksi. Akumulasi rating dikomputasi dari rating yang diberikan pembeli dan penjual. Selanjutnya rating ini disimpan pada server pusat eBay.

Beberapa penelitian telah dilakukan untuk menyimpan dan mendistribusikan informasi reputasi tanpa menggunakan server pusat, salah satunya adalah pada penelitian yang dilakukan oleh Aberer [1]. Penelitian Aberer berfokus pada manajemen dan pencarian nilai kepercayaan dengan menggunakan sebuah database P2P yang terdistribusi. Database ini menyimpan keluhan *peer* yang mempunyai pengalaman transaksi buruk. Ketika agen P ingin mengevaluasi kualitas agen Q, P mengirim kueri untuk mencari data keluhan yang melibatkan Q. Dari data tersebut tingkat kepercayaan terhadap Q dapat diketahui. Protokol ini hanya menggunakan keluhan sebagai parameter yang digunakan untuk mengetahui reputasi *peer*. Sistem ini tidak menggunakan parameter baik atau buruknya transaksi dalam menentukan reputasi sebuah *peer*. Serangan DOS dapat terjadi selama *malicious peer*

dapat memasukkan sejumlah keluhan kedalam sistem tanpa terdeteksi.

## 3. PERANCANGAN

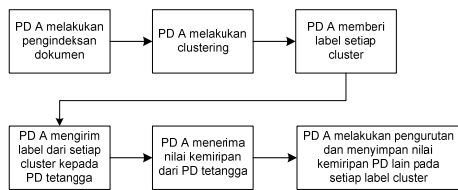
PD berbasis P2P membutuhkan sistem layanan yang dapat memberikan informasi kualitas layanan dari setiap PD. Pada jaringan yang tidak terstruktur, dimana tak ada fungsi kontrol secara terpusat, kehadiran sistem ini menjadi kebutuhan yang penting. PD membutuhkan sistem yang dapat membantu meningkatkan efektifitas dan efisiensi pencarian layanan. Selain itu, PD memerlukan sistem yang dapat menjaga akurasi dan kebenaran nilai QoS dari setiap PD. Lingkungan P2P rawan terhadap keberadaan *malicious peer* yang memberikan nilai QoS rendah bagi PD yang berkualitas baik. Oleh karena itu, kehandalan sistem dalam mengenali PD berkualitas buruk dan PD berkualitas baik sangat dibutuhkan.

Pada penelitian ini, penulis mengusulkan sistem layanan baru bagi PD berbasis P2P. Sistem layanan ini penulis beri nama Jarvis. Secara detail, aktifitas dalam Jarvis dibagi menjadi lima tahapan :

### 1. Inisialisasi kemiripan antar peer

Proses pengelompokan dokumen kedalam sejumlah cluster dilakukan pada saat PD baru bergabung dalam jaringan. Sebelum membentuk cluster, pertama kali PD mengindeks dokumen yang dimilikinya (lihat gambar 1). Pada proses pengindeksan, kata-kata yang masuk dalam kategori *stopwords* dihapus karena belum dapat mencirikan dokumen. Setelah proses pengindeksan selesai, kata-kata yang berhasil diindeks diberi bobot dengan cara membagi jumlah frekuensi kata dengan jumlah semua kata dalam dokumen. Pembobotan ini dilakukan untuk mendukung proses pengelompokan dokumen (*clustering*). Pengelompokan dokumen menggunakan bobot kata yang direpresentasikan dalam bentuk vektor. Selanjutnya proses perhitungan jarak vektor dokumen dan cluster dilakukan dengan menggunakan *cosine similarity*.

Setelah sejumlah cluster terbentuk, proses pendefinisian label yang mencirikan karakter tiap cluster dilakukan. Pemberian label dilakukan dengan cara mengambil sejumlah  $n$  kata yang mempunyai bobot maksimum dari semua dokumen yang tergabung pada sebuah cluster. Label untuk setiap cluster dapat berbeda karena dokumen-dokumen yang tergabung dalam satu cluster mempunyai kemiripan berbeda dengan dokumen-dokumen yang tergabung pada cluster lain.



Gambar 1. Pencarian dokumen

Setelah label terbentuk, selanjutnya PD mengirimkan label dari setiap cluster kepada PD tetangga. PD tetangga melakukan perhitungan *cosine similarity* antara sejumlah cluster yang dimilikinya dengan label cluster yang diterima. Bila PD tetangga mempunyai tiga cluster, maka ketiga cluster diukur jarak kemiripannya dengan setiap label cluster yang diterimanya. Jika nilai kemiripan cluster 3 dengan label 1 lebih tinggi bila dibandingkan dengan cluster 1 dan 2, maka dapat dikatakan label 1 lebih mirip kepada cluster 3 daripada cluster 1 dan 2. Setelah proses perhitungan kemiripan pada PD tetangga selesai, PD tetangga mengirimkan nilai kemiripan dari tiap clusternya kepada alamat PD pengirim. PD yang menerima kiriman nilai kemiripan kemudian menyimpan nilai tersebut kedalam variabel daftar PD. Variabel daftar PD menyimpan kumpulan PD yang terurut pada setiap cluster berdasarkan nilai kemiripan paling besar. Format variabelnya adalah sebagai berikut:

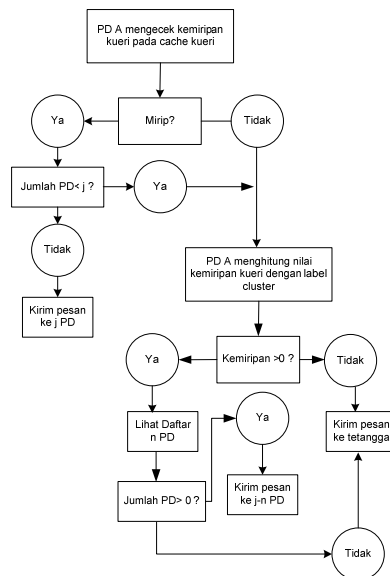
```

$Daftar_PD={ [cluster1-> [PD_E, PD_B]],
              [cluster2-> [PD_C, PD_D]] }
    
```

## 2. Pencarian dokumen

Proses pencarian dokumen dilakukan dengan mengirimkan kueri kepada PD lain. Sebelum pengiriman, terlebih dahulu PD pengirim mengecek rekaman kueri pada *cache* kueri. *Cache* kueri menyimpan daftar PD berdasarkan hasil pencarian dengan kueri sebelumnya. Jika sebelumnya PD telah melakukan kueri yang sama, maka PD akan mengecek daftar PD yang memberi respon terhadap kueri. Untuk kueri yang sama, PD mengirim kueri ke alamat PD dalam daftar *cache* kueri. Jika jumlah PD dalam daftar *cache* kueri belum memenuhi bilangan  $j$ , maka PD akan mengecek daftar PD dari hasil kemiripan antara kueri dengan daftar *cluster*. Nilai  $j$  adalah sama dengan jumlah tetangga yang terhubung dengan PD pengirim.

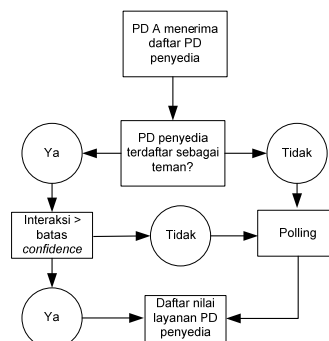
Pada kasus kueri tidak terdaftar dalam daftar *cache* kueri, PD akan menghitung kemiripan kueri dengan label cluster yang dimilikinya (lihat gambar 2). Jika nilai kemiripan antara kueri dengan label cluster 1 lebih besar bila dibandingkan dengan dari label cluster 2 dan 3, maka PD akan mengirimkan pesan kepada sejumlah  $n$  PD yang berada pada cluster 1 dari variabel daftar PD. Jika daftar PD tidak memuat PD, maka pesan dikirimkan secara *broadcast* kepada  $n$  tetangga.



Gambar 2. Pencarian dokumen

## 3. Polling

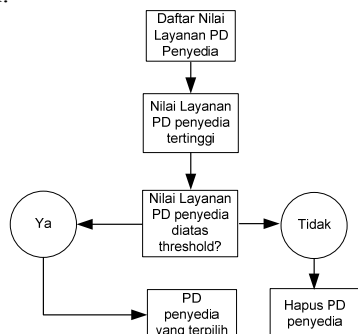
Pada tahap ini, sistem menanyakan nilai QoS sejumlah PD penyedia kepada PD lain. PD lain yang pernah melakukan kerjasama dengan PD penyedia mengirimkan rekaman nilai QoS kepada PD A (PD pengirim). PD yang mengirimkan nilai dapat berjumlah lebih dari satu, tergantung dari banyaknya PD yang pernah bertransaksi dengan PD penyedia tersebut. Bila PD penyedia merupakan teman PD A dan pernah melakukan kerjasama dengan PD A lebih dari batas *confidence*, maka *polling* untuk PD penyedia tersebut tidak perlu dilakukan. Batas *confidence* merupakan batas yang didefinisikan oleh PD sebagai pertanda keyakinan terhadap kualitas layanan PD penyedia. Sebagai contoh bila PD A telah melakukan interaksi dengan PD penyedia (misalnya PD B) lebih dari 4 kali melebihi batas *confidence* (misalnya batas *confidence* =4), maka *polling* tidak diperlukan lagi karena PD A telah yakin terhadap kualitas layanan PD B. Bila ternyata jumlah interaksi PD A dengan PD B belum memenuhi batas *confidence*, maka aktifitas *polling* dilakukan.



Gambar 3. Polling

#### 4. Pemilihan PD penyedia dokumen

Setelah semua nilai hasil *polling* terkumpul, pemilihan PD penyedia dilakukan (lihat gambar 4). Proses pemilihan menggunakan formula matematis dengan menghitung rata-rata nilai hasil polling dari setiap PD penyedia. Sebuah PD penyedia akan terpilih untuk pengambilan dokumen jika PD penyedia memenuhi persyaratan nilai QoS minimum yang ditentukan oleh PD inisiator. Jumlah PD penyedia yang dipilih dapat lebih dari satu mulai dari PD penyedia yang mempunyai nilai rata-rata tertinggi.



Gambar 4. Pemilihan PD penyedia dokumen

#### 5. Koneksi

Pada tahap ini, PD pengirim kueri melakukan koneksi ke PD penyedia yang terpilih pada tahap keempat. Upaya koneksi dilakukan untuk mengunduh dokumen dari PD penyedia. Saat aktifitas koneksi inilah nilai QoS PD penyedia yang sebenarnya diketahui oleh PD pengirim kueri. Nilai QoS ini selanjutnya disimpan oleh PD inisiator dalam penyimpanan lokalnya (*cache*). Jika nilai QoS PD penyedia yang terdaftar dalam variabel daftar PD cluster adalah buruk, maka PD penyedia tersebut dihapus dari daftar. Id PD penyedia yang telah dikunjungi oleh PD pengirim disimpan dalam *cache* kueri.



Gambar 5. Koneksi ke PD penyedia dokumen

#### 4. KONFIGURASI

Penelitian ini menggunakan sejumlah parameter percobaan yang diharapkan dapat mewakili kondisi yang terjadi dilingkungan P2P. Penulis mengikutsertakan variasi sifat PD dalam jaringan yang umum terjadi. Berdasarkan sifat, PD dibagi menjadi tiga kategori:

- PD yang berkualitas baik  
PD jenis ini mempunyai nilai bandwidth, kualitas dokumen, jumlah koleksi dan jumlah interaksi diatas 0.5.
- PD yang berkualitas buruk  
PD jenis ini mempunyai nilai bandwidth, kualitas dokumen, jumlah koleksi dan jumlah interaksi kurang dari 0.5.
- *Malicious PD*

Peer jenis ini mempunyai nilai kualitas dokumen sama dengan nol dan nilai jumlah interaksi yang kurang dari 0.5.

Masing-masing PD dalam jaringan memiliki sebaran jumlah, variasi dan duplikasi dokumen yang berbeda. Pada penelitian, sebaran popularitas kueri diimplementasikan dengan menggunakan distribusi Zipf. Daftar kueri terdiri dari kata-kata yang mempunyai frekuensi tinggi dari kumpulan 10.000 dokumen teks CLEF [4]. Kata-kata yang diambil merupakan kata-kata yang sering muncul dan berada pada ranking 1..10.000 dari total frekuensi kata.

Berikut adalah konfigurasi yang digunakan dalam simulasi:

Tabel 1. Konfigurasi Penelitian

Variabel	Nilai	Keterangan
$\pi$	5000	Jumlah PD dalam jaringan P2P
$\pi_s$	20%	Jumlah PD berkualitas buruk dalam jaringan.
$\pi_b$	15%	Jumlah <i>malicious peer</i> dalam jaringan.
$H$	1	Jumlah batas interaksi dalam nilai <i>confidence</i> n pada formula 4.5
$D$	10.000	Jumlah dokumen unik dalam jaringan
$T_k$	5	Jumlah <i>Time-To-Live</i> pesan kueri
$T_p$	4	Jumlah <i>Time-To-Live</i> pesan <i>polling</i>
$BT$	0.55	Nilai threshold pemilihan teman
$C$	4	Jumlah tetangga yang terhubung ke tiap PD
$zD$	1	Konstanta Zipf untuk distribusi dokumen pada semua PD
$zQ$	1.24	Konstanta Zipf untuk distribusi kueri. Nilai ini diambil berdasarkan penelitian yang dilakukan oleh [14]
$S$	10.000	Waktu akhir simulasi
$Cq$	100	Waktu <i>cycle</i>
$Qt$	500.000	Total kueri
$Qc$	500	Jumlah kueri dalam satu <i>cycle</i>
$I$	0.5	Inisialisasi nilai QoS bagi PD yang belum pernah dikunjungi
$MT$	10	Jumlah kata maksimum pada satu label cluster
$Ct$	4	Jumlah cluster dokumen pada masing-masing peer
$N$	30	Jumlah dokumen yang diinginkan pada saat kueri

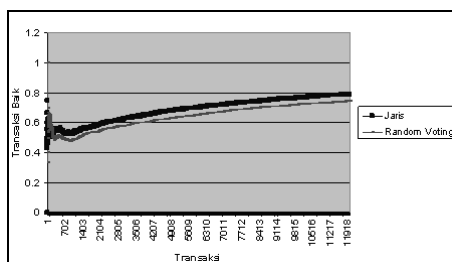
Variabel	Nilai	Keterangan
<i>M</i>	2500	Jumlah maksimum indeks yang disimpan oleh <i>cache</i> kueri
<i>M<sub>p</sub></i>	4	Jumlah maksimum daftar PD yang disimpan untuk setiap kueri dalam <i>cache</i> kueri
<i>C<sub>s</sub></i>	4	Jumlah maksimum PD referer yang disimpan pada setiap cluster

## 5. HASIL PERCOBAAN DAN ANALISA

Pada penelitian ini, evaluasi dilakukan pada empat aspek yaitu akurasi transaksi, jumlah *hop*, kepadatan lalu lintas data dan kemiripan kueri-dokumen. Kueri diambil dari kata-kata populer seluruh dokumen yang dimiliki PD. Pengiriman kueri pada skenario ini mengikuti sebaran distribusi Zipf.

### 5.1 Akurasi Transaksi

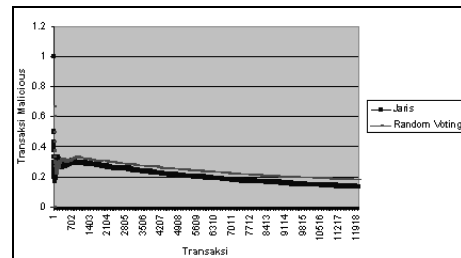
*Perbandingan jumlah kejadian transaksi PD berkualitas baik*



Gambar 6. Persentase jumlah transaksi baik

Pada grafik gambar 6, nilai akurasi transaksi *random voting* lebih rendah bila dibandingkan dengan nilai akurasi metode jaris. Fenomena ini dapat terjadi karena metode *random voting* melakukan polling secara *broadcast* tanpa melalui jaringan pertemanan PD. Hal ini menyebabkan *malicious* PD berpeluang untuk memberikan respon palsu terhadap kualitas layanan PD yang ditanyakan. Selain itu *random voting* hanya menggunakan nilai polling global sebagai penentu keputusan. Nilai polling global merupakan nilai mengenai suatu PD *x* yang didapat dari PD lain. Padahal kenyataannya nilai polling global rawan terhadap nilai QoS palsu dari *malicious* PD.

*Perbandingan jumlah transaksi dengan malicious peer*

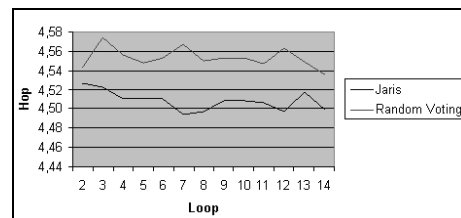


Gambar 7. Persentase jumlah transaksi dengan *malicious* PD

Pada gambar 7, keunggulan jaris terhadap metode *random voting* hanya terpaut pada skala 4-7 persen. Penurunan poin yang lebih cepat dari metode jaris kemungkinan besar tidak lepas dari peranan penggunaan rekaman nilai QoS lokal dalam perhitungan pemilihan PD penyedia. Nilai QoS lokal merupakan nilai QoS yang didapat berdasarkan pengalaman transaksi PD pengirim dengan PD penyedia. PD penyedia merupakan PD yang mempunyai dokumen diinginkan PD pengirim. Pada metode jaris, PD pengirim mengkombinasikan perhitungan rekaman nilai QoS lokal dengan nilai QoS global untuk memilih PD penyedia. Pada batas interaksi tertentu, sebuah PD hanya memperhitungkan nilai QoS lokal tanpa memasukkan nilai QoS global dalam pemilihan PD penyedia.

### 5.2 Hop

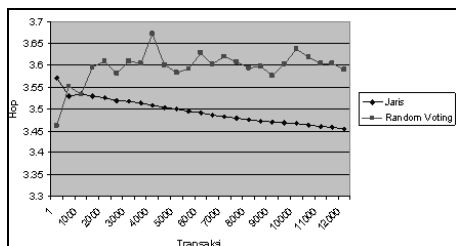
*Hop untuk pencarian dokumen*



Gambar 8. Perbandingan rata-rata *hop*

Metode *random voting* menggunakan metode *broadcast* untuk mencari dokumen yang relevan. Pencarian ini mempunyai batasan yang sama dengan metode jaris. Salah satu perbedaan *random voting* dengan jaris adalah pada pencarian dokumen PD tanpa menggunakan informasi kemiripan dan *cache* kueri. Performa *random voting* dapat dilihat pada gambar 8. Pada grafik jumlah rata-rata *hop random voting* pada tiap transaksi stabil pada kisaran 4.54 - 4.57.

Hop untuk kueri polling

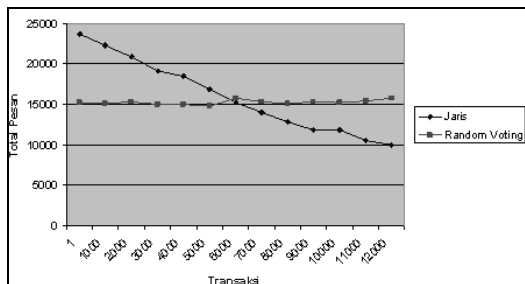


Gambar 9. Perbandingan hop polling

Gambar 8 dan 9 merepresentasikan rataan hop polling terjadi pada setiap transaksi. Bila dibandingkan dengan random voting, rata-rata hop polling jaris lebih rendah walaupun pada saat awal transaksi hop polling jaris lebih tinggi. Hal ini dapat terjadi karena pada saat awal polling jaris belum mempunyai rekaman nilai layanan dari PD lain. Saat jumlah transaksi semakin banyak dan rekaman nilai semakin bertambah, maka ketika terdapat sejumlah PD penyedia yang telah diketahui nilai QoS nya, mekanisme polling tidak perlu dilakukan kembali. Hal ini menyebabkan rata-rata hop polling semakin sedikit seiring dengan peningkatan jumlah transaksi.

5.3 Kepadatan lalu lintas data

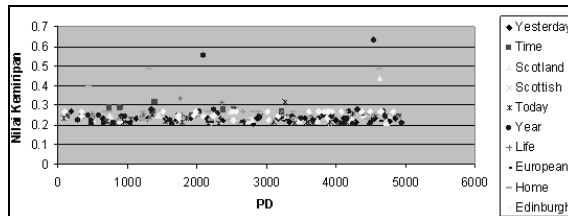
Pada uji coba kepadatan lalu lintas data, total pesan yang dikirim oleh metode jaris lebih sedikit bila dibandingkan dengan metode random voting seiring dengan peningkatan jumlah transaksi. Walau pada awal transaksi total pesan yang dikirim jaris lebih besar bila dibandingkan dengan metode random voting, namun seiring dengan peningkatan jumlah transaksi, total pesan jaris perlahan semakin turun. Kemungkinan besar fenomena ini terjadi karena rendahnya relevansi antara kueri dengan cache kueri dan informasi kemiripan dalam cluster saat awal transaksi. Pada kenyataannya cache kueri sangat tergantung dari banyaknya transaksi yang dilakukan sebuah PD. Semakin banyak transaksi, maka cache kueri akan semakin besar sehingga kemungkinan munculnya kueri yang relevan dengan cache kueri tinggi.



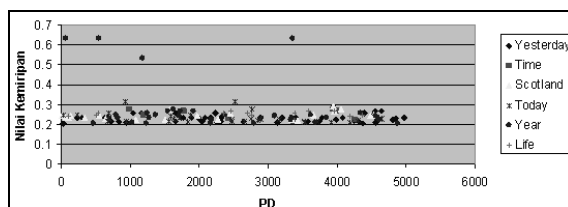
Gambar 10. Perbandingan total pesan

5.4 Kemiripan Kueri

Berdasarkan hasil ujicoba, umumnya nilai kemiripan antara kueri dan dokumen berada pada skala 0.2 – 0.3 pada threshold 0.2. Walau begitu ada beberapa dokumen yang mempunyai nilai kemiripan diatas skala 0.2-0.3, namun jumlahnya kecil sekali. jumlah PD yang memberi respon pada kedua metode cenderung fluktuatif. Kemungkinan besari fluktuasi ini terjadi karena perbedaan popularitas kueri yang dikirimkan.



Gambar 11. Perbandingan kemiripan kueri pada Jaris



Gambar 12. Perbandingan kemiripan kueri pada Random voting

Secara umum, semakin populer kueri yang dikirim, maka jumlah PD yang memberi respon semakin banyak. Nilai kemiripan dari kedua kueri juga tidak lepas dari tingginya bobot kata kueri pada dokumen yang dimiliki PD. Semakin tinggi bobot kata kueri pada dokumen, maka nilai kemiripan kueri-dokumen semakin tinggi. Begitu juga sebaliknya, semakin rendah bobot kata kueri pada dokumen yang dimiliki PD, maka nilai kemiripan kueri-dokumen semakin rendah.

6. KESIMPULAN

Rangkaian pengujian terhadap kinerja metode jaris telah dilakukan. Pengujian mengacu pada aspek akurasi transaksi, jumlah hop, kepadatan lalu lintas data dan kemiripan kueri-dokumen. Keempat aspek tersebut diujikan melalui simulasi penelitian. Simulasi juga mengikutsertakan metode lain yaitu metode random voting. Pada penelitian, kinerja metode random voting dibandingkan dengan metode jaris. Dari hasil simulasi dapat ditarik kesimpulan bahwa secara umum kinerja metode jaris lebih baik bila dibandingkan dengan metode random voting. Hal ini dapat dilihat dari perbandingan jumlah transaksi dengan PD berkualitas baik dan jumlah transaksi dengan PD yang bersifat malicious. Jumlah kejadian transaksi dengan PD berkualitas baik unggul pada skala 3-10 persen sedangkan pada jumlah kejadian transaksi dengan malicious peer, jaris dapat menekan hingga 4-7 persen kejadian.

Selain itu dari segi jumlah *hop* dan jumlah pesan, jumlah *hop* yang dihasilkan oleh metode Jaris lebih rendah 0.1- 0.11 poin dan jumlah pesan yang dihasilkan semakin sedikit seiring dengan peningkatan jumlah transaksi. Jaris menggungguli penurunan jumlah pesan pada saat transaksi ke-6000 pada skenario satu dan transaksi ke-7500 pada skenario dua.

#### PUSTAKA

- [1] Aberer, Karl., Despotovic, Zoran. 2001. *Managing trust in a peer-2-peer information system*. In Henrique Paques, Ling Liu, and David Grossman, editors, Tenth International Conference on Information and Knowledge Management (CIKM01), pages 310–317. ACM Press, 2001
- [2] Amrou, A., Maly, K., Zubair, M. 2006. *Freelib: Peer-to-peer-based Digital Libraries*. Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06) - Volume 01, Pages: 9 – 14
- [3] Aringhieri, Roberto., Bonomi, Daniele. 2004. *A Simulation Model for Trust and Reputation System Evaluation in a P2P Network*
- [4] CLEF. 2007. <http://www.clef-campaign.org/>
- [5] Cornelli, F., Damiani, E., Vimercati, S. De Capitani di., Paraboschi S., Samarati, P. 2002. *Choosing Reputable Servents in a P2P Network*. in *Proc. of the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, May 7-11
- [6] Damiani, Ernesto., Vimercati, De Capitani di., Paraboschi, Stefano., Samarati, Pierangela., Violante, Fabio. 2002. *A reputation-based approach for choosing reliable resources in peer-to-peer networks*. In Ninth ACM conference on Computer and communications security, pages 207–216. ACM Press
- [7] eBay. 2007. <http://www.ebay.com>
- [8] Kwok, S.H., Chan, K.Y., Cheung, Y. M. 2005. *A Server-mediated Peer-to-peer System*, ACM SIGecom Exchanges, Vol. 5, No. 3, April 2005, Pages 38–47.
- [9] Marti, Sergio. 2005. *Trust And Reputation In Peer-To-Peer Networks*. A Dissertation Submitted To The Department Of Computer Science And The Committee On Graduate Studies Of Stanford University
- [10] Napster. 2007. <http://www.napster.com>
- [11] Oram, Andy. 2001. *Peer to Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly & Associates, Inc., First Edition
- [12] Risse, Thomas Risse., Knežević, Predrag., Meghini, Carlo., Hecht, Robert., and Basile, Fiore. 2006. *The BRICKS Infrastructure - An Overview*
- [13] Salton, G., Wong, A., Yang, C. S. 1975. *A Vector Space Model for Automatic Indexing*. Communications of the ACM, vol. 18, nr. 11, pages 613–620.
- [14] Sripanidkulchai, K., Maggs, B., Zhang, H. 2003. *Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems*. INFOCOM
- [15] Walkerdine, J. and Rayson, P. 2004. *P2P-4-DL: Digital Library over Peer-to-Peer*. In Caronni G., Weiler N., Shahmehri N. (eds.) Proceedings of Fourth IEEE International Conference on Peer-to-Peer Computing (PSP2004) 25-27 August 2004, Zurich, Switzerland. IEEE Computer Society Press, pp. 264-265. ISBN 0-7695-2156-8
- [16] Xu, Yanfei. 2005. *A P2P Based Personal Digital Library for Community*. Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on volume , Issue , 05-08 Dec. 2005 Page(s): 796 – 800