

Implementasi Algoritma Gosip Dalam Jaringan *Peer-to-Peer*

Aditya Nugraha Tama¹, Saptadi Nugroho², Hartanto Kusuma Wardana³

Program Studi Sistem Komputer,
Fakultas Teknik Elektronika dan Komputer,
Universitas Kristen Satya Wacana, Salatiga
¹622013002@student.uksw.edu, ²saptadi.nugroho@staff.uksw.edu,
³hartanto.kusuma@staff.uksw.edu

Ringkasan

Komputer merupakan perangkat yang dapat melakukan banyak hal, termasuk berkomunikasi satu sama lain. Seperti halnya manusia dengan pengaturan yang tepat komputer bisa mendapatkan suatu informasi terkait suatu data melalui proses komunikasi. Oleh sebab itu, diperlukan sebuah protokol jaringan yang dapat mengatur bagaimana komputer dapat terhubung satu sama lain serta mengatur proses pertukaran data. Algoritma gosip merupakan algoritma yang mengatur proses pertukaran data antar dua buah *node* yang terhubung. Untuk mengimplementasikan algoritma gosip terdapat dua hal yang paling utama yaitu *active thread* dan *passive thread*. *Active thread* diawali saat proses menghubungi *node* tetangga untuk diajak bertukar data dijalankan, sedangkan *passive thread* akan menunggu sampai ada ajakan dari tetangga. Pengujian yang telah dilakukan dengan dua buah *node* adalah pengujian proses komunikasi antar *node* dan proses pertukaran serta *update* data. Dari hasil pengujian, proses komunikasi dan pertukaran serta *update* data dapat berjalan baik dan lancar.

Kata kunci: algoritma gosip, protokol, jaringan

1. Pendahuluan

Kemajuan teknologi di bidang komputer yang semakin pesat, membuat banyak hal dalam kehidupan menjadi serba mudah. Komputer dapat membantu bahkan mengerjakan tugas manusia seperti, mengolah serta menganalisis data, mengendalikan serta mengatur proses produksi, dan terhubung dengan milyaran komputer lain yang tersebar di seluruh dunia melalui jaringan internet. Melalui internet, komputer dapat melakukan pertukaran data dengan memenuhi suatu protokol komunikasi yang ada. Dengan kata lain, komputer satu dengan komputer lain dapat melakukan proses komunikasi layaknya manusia.

Dalam kehidupan, manusia pasti melakukan proses komunikasi dengan orang lain. Salah satu bentuk komunikasi yang dilakukan oleh manusia adalah bergosip. Gosip merupakan salah satu cara komunikasi yang digemari manusia, dimana telah dilakukan sejak jaman Romawi sampai dengan sekarang [1]. Ketika manusia bergosip akan terjadi proses pertukaran informasi, sehingga bisa didapatkan suatu informasi baru yang belum diketahuinya. Walaupun informasi yang didapatkan belum tentu kebenarannya, gosip dapat menyebar dengan cepat. Hal inilah yang kemudian memicu para ilmuwan untuk membuat komputer dapat menyebarkan informasi layaknya manusia bergosip.

2. Dasar Teori

2.1. Algoritma Gosip

Awal mula dari algoritma gosip adalah untuk menghitung jumlah panggilan telepon yang dibutuhkan untuk membuat semua wanita mengetahui gosip yang sama, ketika terdapat sejumlah n wanita yang mengetahui suatu informasi yang tidak diketahui oleh wanita lain. Mereka berkomunikasi melalui telepon, dan saat seorang wanita menelpon wanita lain mereka akan saling bertukar informasi [2]. Berbeda dengan algoritma *flooding* yang murni, gosip mengurangi jumlah operasi per *node* per satuan waktu, mengurangi konsumsi daya per *node*, dan jumlah total transmisi dalam satuan waktu (mengurangi interferensi), yang membuatnya sangat sesuai bila digunakan untuk suatu proses komunikasi jaringan [3][4]. Algoritma gosip merupakan salah satu metode komunikasi kuno yang penting pada jaringan yang besar, dan telah dipelajari dengan berbagai macam istilah lain seperti *rumor spreading*, *information dissemination*, dan *broadcasting* [5].

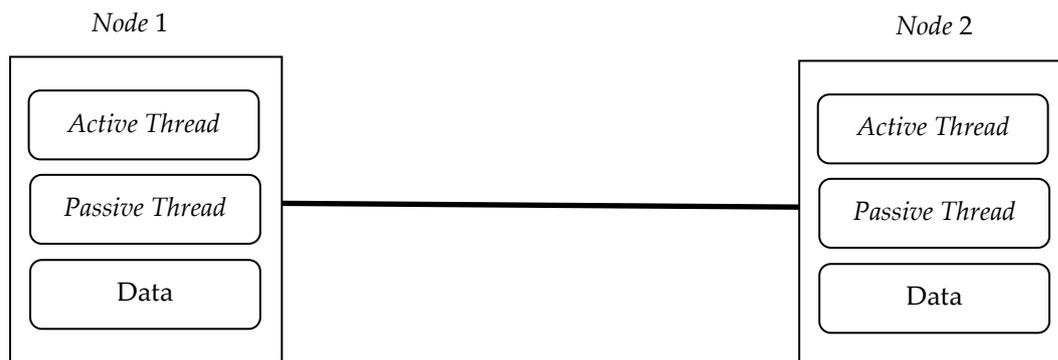
2.2. Peer-to-Peer

Jaringan *peer-to-peer* (P2P) adalah jaringan dimana semua node dapat bertindak sebagai client/server, karena saling terhubung secara langsung satu sama lain. Selain itu, dalam jaringan P2P tidak terdapat titik pengaturan yang terpusat. Daya tarik utama dari jaringan P2P adalah dapat berbagi sumber daya terdistribusi, sehingga menghindari duplikasi dan biaya tambahan. Satu atau lebih komputer dapat berbagi *file*, *printer*, *drive* optik dan sumber daya lain [6].

3. Perancangan

3.1. Gambaran Sistem

Pada setiap *node* terdapat tiga hal penting yaitu *active thread*, *passive thread*, serta data yang digunakan dalam proses komunikasi.



Gambar 1. Gambaran Sistem

Node akan secara otomatis menjalankan *passive thread* sehingga setiap *node* selalu menunggu dan mengecek apakah ada kiriman dari *node* lain, ketika terdapat kiriman maka *node* akan membalas kiriman tersebut dengan data miliknya. Sedangkan *active thread* dijalankan ketika *node* memulai proses pertukaran data, yaitu ketika data dikirim ke *node* tetangga[7].

3.2. *Active Thread*

Active thread diawali dengan menunggu selama selang waktu yang ditentukan, kemudian menjalankan fungsi pemilihan tetangga. Karena hanya terdapat dua buah *node*, maka IP *address* tetangga dapat dimasukkan secara manual. Kemudian, data akan dikirimkan kepada IP tersebut. Ketika ada data balasan dari tetangga, maka terima data, kemudian hitung rata-rata dan *update* data. Jika tetangga tidak membalas, maka ulangi proses dari awal. Pada sistem ini *active thread* bekerja secara terus-menerus (perulangan tak terbatas).

3.3. *Passive Thread*

Saat *passive thread* dijalankan, maka *node* akan menunggu untuk dihubungi oleh *node* lain. Jika ada *node* yang menghubungi dan kiriman data diterima, kemudian *node* tersebut akan menyimpan kiriman data tersebut, lalu akan mengirim datanya sebagai balasan. Kemudian, rata-rata akan dihitung dari penjumlahan dari data yang diterima dengan data sendiri dibagi dengan dua. Setelah menghitung rata-rata, maka data dari *node* akan di *update*.

3.4. Realisasi Sistem

Untuk merealisasikan sistem dibutuhkan alat-alat berupa Raspberry Pi sebagai *node* dan kabel *ethernet* untuk meyambungkan 2 buah *node*. *Prefix* yang digunakan dalam jaringan ini adalah /24. Sehingga dalam satu jaringan terdapat 254 *host* yang dapat digunakan. Jaringan yang digunakan berada pada kelas C. Pemberian IP *address* dalam sistem ditunjukkan dalam Tabel 1.

Tabel 1. Pembagian IP *Address* Pada Sistem

Nama	<i>Network ID</i>	IP <i>Address</i>
<i>Node 1</i>	192.168.30.0 /24	192.168.30.2
<i>Node 2</i>	192.168.30.0 /24	192.168.30.20

4. Pengujian dan Analisis

Dari hasil implementasi maka dapat dilakukan proses pengujian pada proses komunikasi yang terbentuk antara dua buah *node* dan pertukaran serta *update* data. Data awal dari *node 1* bernilai 10, sedangkan *node 2* bernilai 20. Pada Gambar 2. Terlihat bahwa *node 1* mengirimkan data awalnya yang bernilai 10 kepada *node 2* yang memiliki IP *address* 192.168.30.20. Kemudian dari Gambar 3. Terlihat bahwa *node 2* telah menerima data yang bernilai 10, lalu data tersebut disimpan. Kemudian, *node 2* membalas dengan data awal miliknya yang bernilai 20. Proses selanjutnya yang terjadi pada *node 2* adalah menghitung rata-rata dari data yang diterima dengan data miliknya yang akan digunakan sebagai data yang dikirimkan pada proses pengiriman selanjutnya. Kembali pada *node 1* setelah menerima balasan data yang bernilai 20 dari *node 2* maka proses selanjutnya adalah menghitung rata-rata dari data yang diterima dengan data miliknya yang akan digunakan sebagai data yang dikirimkan pada proses pengiriman selanjutnya. Sehingga kedua *node* sepakat terhadap bahwa data bernilai 15.

```

pi@raspberrypi: ~/Desktop/GOSIP
File Edit Tabs Help
pi@raspberrypi:~/Desktop/GOSIP $ sudo python ActiveThread.py
dikirim ke ('192.168.30.20', 8888) > 10
balasan dari ('192.168.30.20', 8888) > 20
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0
dikirim ke ('192.168.30.20', 8888) > 15.0
balasan dari ('192.168.30.20', 8888) > 15.0

```

Gambar 2. Tampilan *Active Thread Node 1*

```

pi@raspberrypi:~/Desktop/GOSIP $ sudo python PassiveThread.py
terima dari ('192.168.30.2', 9999) > 10
dibalas > 20
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0
terima dari ('192.168.30.2', 9999) > 15.0
dibalas > 15.0

```

Gambar 3. Tampilan *Passive Thread Node 2*

5. Kesimpulan

Implementasi algoritma gosip dapat dilakukan pada dua buah Raspberry Pi yang berfungsi sebagai *node* yang terhubung secara *peer-to-peer*. *Node* satu dengan yang lain dapat saling berkomunikasi untuk bertukar data. Proses pengiriman dan pembaruan data pada tiap *node* berjalan dengan baik. Hal ini dibuktikan dengan data yang bernilai sama saat *node* saling terhubung.

Daftar Pustaka

[1] M. Crescimbene, F. Longa, and T. Lanza, *The Science of Rumors*, Roma: Annals of Geophysics, h. 421-425, 2012.

- [2] B. Baker and R. Shostak, *Gossip and Telephones*, North Holland Publishing Company, h. 191-193, 1971.
- [3] S. Boyd, Arpita Ghosh, et al., "Analysis and optimization of randomized gossip algorithms", *IEEE Trans. on Inf. Theory*, Vol. 52, No. 6, 2004.
- [4] L. Alvisi, et al., "How robust are gossip-based communication protocols?" *ACM SIGOPS Operating Systems Review*, h. 14-18, 2007.
- [5] Z. Guo and H. Sun, "Gossip vs. Markov Chains, and Randomness-Efficient Rumor Spreading," *SODA '15 Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, h. 411-430, 2015.
- [6] B. Sossinsky, *Networking Bible*, Indiana: Wiley Publishing Inc., 2009.
- [7] R. Bakhshi, D. Gavidia, et al., "A modeling framework for gossip-based information spread", *In Quantitative Evaluation of Systems (QEST) 2011 Eighth International Conference*, h 245–254, 2011.

