

# Rancang Bangun Aplikasi Sosial Media Crawler Menggunakan Nodejs Menerapkan Konsep Non-Blocking I/O

Chandra Yuda Andika<sup>1</sup>, Sabar Rudiarto<sup>2</sup>

<sup>12</sup>Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana

<sup>1</sup>Jalan Meruya Selatan 1 No.1, Kembangan, Jakarta Barat 11650

e-mail : <sup>1</sup>chandra.yuda.andika@gmail.com, <sup>2</sup>sabar.rudiarto@mercubuana.ac.id

## ABSTRAK

*Semakin majunya dunia teknologi dan informasi memberikan dampak besar bagi kehidupan kita, misal saja kemudahan akses komunikasi, berbicara pada orang lain cukup dengan benda kecil yang bisa kita masukan dalam saku celana, menyebarkan informasi ?, cukup dengan aplikasi – aplikasi yang tersedia, baik itu pada perangkat komputer maupun mobile, pada dasarnya informasi yang kita sebarkan memiliki nilai yang bisa dimanfaatkan melalui suatu analisa terlebih dahulu, namun saat ini pengambilan data data yang telah tersebar tersebut masih cukup sulit, karena saat ini aplikasi pengambilan data prosesnya hanya dibuat pada satu layer yang mana prosesnya menerapkan antrian satu jalur, oleh karena itu diperlukan adanya aplikasi yang menerapkan konsep Non-blocking I/O pada pembangunan dan perancangan aplikasi bisa menjadi suatu solusi.*

**Kata kunci:** Layer, Media Sosial, Non-blocking I/O

## 1. PENDAHULUAN

Semakin majunya dunia teknologi dan informasi memberikan dampak besar bagi kehidupan kita, misal saja kemudahan akses komunikasi, berbicara pada orang lain cukup dengan benda kecil yang bisa kita masukan dalam saku celana, menyebarkan informasi?, cukup dengan aplikasi – aplikasi yang tersedia, baik itu pada perangkat komputer maupun mobile (bergerak), pada penelitian ini penulis membahas mengenai pemanfaatan data – data yang telah kita sebarkan pada aplikasi – aplikasi yang tersedia, misal saja, berapa jumlah status yang diupdate pada media sosial Facebook, berapa banyak twit yang ada pada Twitter, berapa banyak lagu yang ada pada Spotify, berapa banyak video yang ada pada Youtube?, saat ini khalayak hanya berperan sebagai pelanggan yang mana hanya menggunakan fitur – fitur yang tersedia pada aplikasi – aplikasi tersebut, dimana setelah sekian lama, informasi itu hanya akan menjadi sampah digital, hal ini sangat disayangkan dimana data yang seharusnya dapat menghasilkan informasi yang berharga dibiarkan begitu saja[9], lalu bagaimanakah cara pemanfaatan yang penulis maksud?, dalam hal ini penulis akan membatasi ruang lingkup media sosial yang digunakan pada Twitter.

Data Twitter atau yang biasa disebut sebagai twit adalah salah satu bentuk informasi yang kita sebarkan melalui media sosial Twitter[1], dengan metode dan analisa yang tepat, setiap twit bisa menghasilkan suatu data yang sangat berguna[12], misal saja berapa banyak orang di Jakarta yang membicarakan tentang pendidikan, atau topik apa yang paling populer di Jakarta, informasi tersebut bisa kita manfaatkan sebagai data pendukung dalam suatu pengambilan keputusan[11].

Pemilihan Twitter sebagai sumber data penelitian dikarenakan Twitter memiliki suatu perantara –API ( Application Programming Interface ) yang memungkinkan user dan pengembang untuk mengakses data publik yang ada pada server Twitter. Seperti twit user, follower, lokasi twit dan sebagainya. Selain itu Twitter juga sudah banyak digunakan sebagai data pebantu dalam pengambilan keputusan di berbagai bidang[11].

Analisa data Twitter tidak terlepas dari metode atau cara pengambilan data yang ada di internet, namun metode metode yang tersedia saat ini belum efektif dan efisien oleh karenanya guna memperoleh data twit dengan jumlah besar secara efektif dan efisien perlu adanya kolaborasi beberapa teknologi dan konsep yang sudah tersedia, antara lain teknologi Nodejs dengan konsep Non Blocking I/O.

Teknik pengambilan data yang bisa digunakan ada 2 crawling dan scraping, Crawling adalah proses penjelajahan suatu konten atau informasi digital yang lalu diunduh ke media penyimpanan, pelaku crawling disebut crawler. Sedangkan scraping adalah proses pembacaan sumber daya suatu halaman web yang kemudian disimpan pada suatu media penyimpanan, pelaku scraping disebut scraper. Ke dua hal tersebut biasa dilakukan sebagai awal proses data mining atau analisa data. Pengambilan dan analisa informasi dilakukan oleh data scientist yang biasanya akan disajikan sebagai report maupun ditampilkan menjadi dashboard sebagai pertimbangan pengambilan keputusan oleh user kedepannya. Berdasarkan uraian di atas, dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana pemanfaatan teknologi non-blocking I/O pada software crawler agar pengambilan data digital lebih efektif dan efisien?
2. Bagaimana cari implementasi teknologi non-blocking I/O pada software crawler agar pengambilan data digital lebih efektif dan efisien?

Agar penelitian rancang bangun aplikasi sosial media crawler menggunakan Node JS menerapkan konsep non-blocking I/O tidak menyimpang dari rumusan permasalahan maka perlu adanya batasan-batasan, antara lain:

1. Dalam pembuatan aplikasi terbatas pada media sosial Twitter.
2. Object informasi *digital* berupa konsumsi publik berasal dari media sosial Twitter.
3. Data hasil *crawling* akan ditampilkan dalam bentuk raw data (JSON).
4. Aplikasi Sosial Media Crawler menggunakan Node JS memanfaatkan non-blocking I/O pada dasarnya berjalan disemua platform namun dalam pengujian hanya akan dijalankan pada platform berbasis windows.
5. Pengujian menggunakan satu buah *server/node*.
6. Aplikasi dibuat menggunakan Node JS.

## 2. LANDASAN TEORI

### 2.1 Aplikasi Crawler / Crawling

Aplikasi crawling adalah suatu aplikasi yang digunakan untuk pengambilan data digital secara mendalam, berbeda dengan scraper yang hanya mengambil data pada permukaan saja, crawler umum digunakan oleh web mesin pencari seperti Google, Yahoo dan sebagainya, yang bertujuan untuk memetakan konten/isi pada suatu halaman web dan dipetakan pada index, dalam lingkup aplikasi yang penulis buat, crawler bertujuan untuk mengambil data secara sistematis, efektif serta efisien pada Twitter melalui API (*Application Programming Interface*) yang sudah disediakan oleh Twitter[7].

### 2.2 Twitter

Twitter adalah layanan jejaring sosial dan mikroblog online yang memungkinkan penggunaanya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter, yang dikenal dengan sebutan kicauan (tweet / twit). Twitter didirikan pada bulan Maret 2006 oleh Jack Dorsey, dan situs jejaring sosialnya diluncurkan pada bulan Juli. Sejak diluncurkan, Twitter telah menjadi salah satu dari sepuluh situs yang paling sering dikunjungi di Internet.

### 2.3 Non-blocking I/O

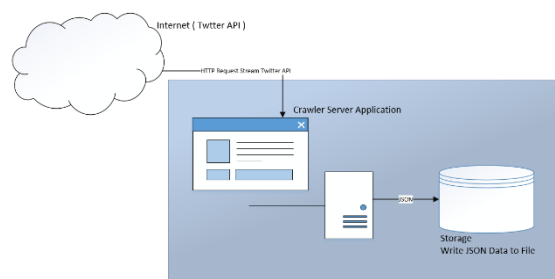
Non-Blocking I/O processing atau Asynchronous processing atau parallel processing adalah suatu cara manipulasi dan pengendalian suatu antrian proses pada suatu layer aplikasi, saat ini sebuah processor computer telah dibuat menjadi bagian bagian kompleks yang menerapkan konsep MPSoC ( Multi Processor System-on-a Chip )[10], dimana kita dapat menjalankan berbagai proses dalam satu waktu. Dengan demikian suatu proses tidak akan menjadi hambatan untuk proses lainnya, karena proses lain bisa di-handle oleh unit pemroses lain, dengan demikian pembuatan aplikasi dengan menerapkan konsep Non-blocking I/O menjadi lebih efektif & efisien karena akan mengurangi jumlah waktu eksekusi [6,3].

### 2.4 Node js

Node js adalah platform yang dibangun di atas runtime Javascript yang ada pada chrome yang berguna untuk proses development aplikasi secara cepat dan efisien serta mempermudah pembangunan aplikasi berbasis jaringan yang memiliki scalability ( daya pengembangan ) yang tinggi, Node Js menggunakan event-driven non-blocking I/O model yang membuat Node Js ringan dan efisien, cocok untuk aplikasi data-intensive realtime yang berjalan pada cross platform[2].

### 2.5 Perancangan

Guna memecahkan masalah yang diuraikan di atas maka dibuatlah aplikasi Sosial Media Crawler ( Twitter ) Menggunakan Node JS Memanfaatkan Non-Blocking I/O dengan arsitektur sebagaimana digambarkan pada Gambar 1.



Gambar 1 Arsitektur Aplikasi

1. Application, Application Server & Storage

Aplikasi berjalan pada sebuah server dengan storage untuk menyimpan data , aplikasi mengambil data yang berada di server Twitter dengan penggunaan HTTP Request API yang telah di sediakan oleh Twitter,

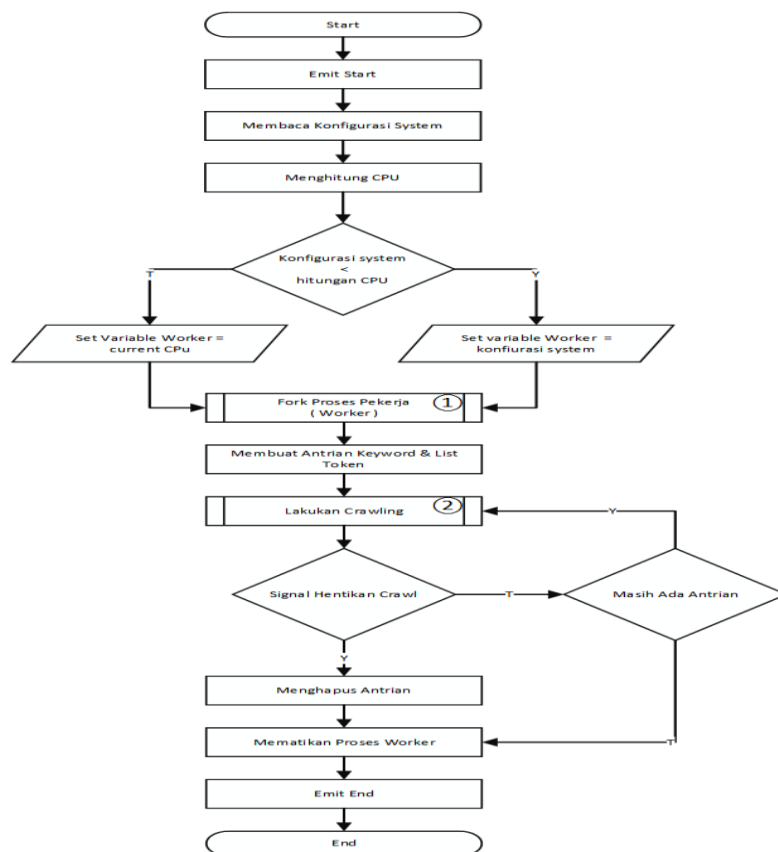
kemudian data ditampilkan di aplikasi (memory) yang lalu write stream ke file dengan format JSON (*Javascript Object Notation*).

2. Twitter Public API  
Adalah suatu *interface* yang disediakan oleh Twitter bagi user dan pengembang guna mendapatkan data publik Twitter [1].

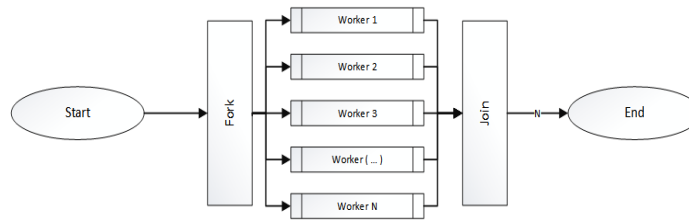
Alur proses *crawl* aplikasi:

1. Aplikasi memulai proses *crawl*
2. Aplikasi membaca konfigurasi CPU atau unit promproses pada sistem aplikasi yang telah di-*setting*  
Aplikasi membaca & menghitung konfigurasi CPU pada sistem *node / server*
3. Melakukan pengecekan, apakah jumlah CPU pada sistem aplikasi lebih banyak dari pada jumlah CPU yang ada pada *node / server*
  - a. YA : Variabel *worker* akan di-*set* sesuai konfigurasi aplikasi, setting dimasukan dari menu Setting > general setting
  - b. TIDAK : Variabel *worker* akan di-*set* sesuai dengan jumlah CPU pada *node / server*
4. Fork proses pekerja ( *worker* ) sesuai dengan variabel *worker*
5. Membuat antrian *keyword* dan antrian token berdasarkan parameter yang sudah di input pada masing masing menu “Keywords” dan “Tokens”
6. Melakukan pendaftaran proses *crawler* pada masing masing *worker* yang *idle*
7. Melakukan pengecekan, adanya signal hentikan *crawling*?
  - a. YA : Menghapus antrian
  - b. TIDAK : Melanjutkan *crawling*
8. Melakukan pengecekan, adanya antrian dalam *pool*?
  - a. YA : Kembali proses nomor 7
  - b. TIDAK : Menghapus antrian
9. Matikan proses *worker*

Alur dan proses *forking* dari sub proses 1 ( Gambar 2 ) Fork Proses Pekerja, pada tahap ini main proses akan mengirim perintah untuk melakukan *forking worker* proses sebanyak variabel *worker*. Menggunakan perintah `require('child_process').fork()` pada perulangan CPU dijelaskan pada Gambar 3.

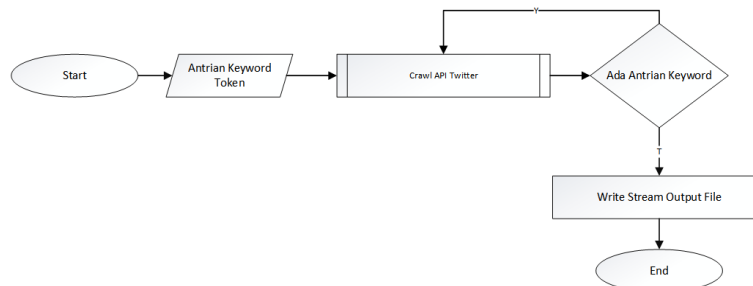


Gambar 2 Alur Crawl Menu “Process”



Gambar 3 Proses Forking Child

Worker Proses (Gambar 4), pada proses ini worker melakukan koneksi terhadap server API Twitter dengan mengirimkan parameter antrian keyword & akses token, akses token adalah suatu parameter autentikasi yang digunakan untuk mendapatkan akses OAuth ke data publik Twitter dengan berates namakan user, dimana kita harus terlebih dahulu mendaftarkan diri yang lalu membuat suatu aplikasi twitter. Yang kemudian Twitter akan memberikan response data, aplikasi membuat system emitter untuk listening terhadap suatu emitter yang dikirimkan oleh proses crawling, Output berupa data dengan format JSON menggunakan metode Write Stream dimana data ditulis kedalam storage secara parsial membentuk chunk – chunk atau potongan data dengan format binary buffer yang kemudian di encode utf-8.

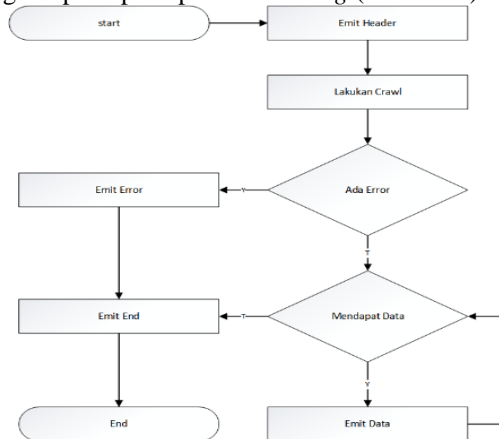


Gambar 4 Proses Crawler

Untuk menjabatani proses yang worker maka dibuatlah sekumpulan emitter, yang akan memberikan serangkaian callback guna meneruskan proses berikutnya, emitter yang digunakan:

- Error : emitter yang akan menerima listening ketika terjadi error, berisi detail error yang secara otomatis di generate oleh system,
- Info : emitter yang akan menerima listening event info
- Header : emitter yang akan menerima listing event header, berisi detail keyword yang akan dicrawling saat itu.
- Data : emitter yang akan menerima listening event header, berisi pesan alur proses pada saat crawler berjalan
- End : emitter yang akan menerima listening event end, berisi pesan informasi bahwa proses crawling sudah selesai, yang artinya crawler sudah tidak mendapat data lagi.

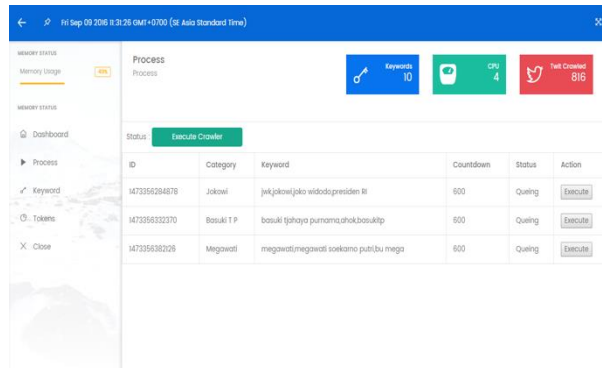
Alur kerja sistem emitter yang terapkan pada proses crawling (Gambar 5)



Gambar 5 Emitter Proses Crawler

3. HASIL DAN PEMBAHASAN

Hasil rancangan aplikasi tersebut akan menjadi dasar dalam tahap pengembangan aplikasi, diwujudkan dalam bentuk aplikasi sosial media crawler berbasis *desktop* yang mendukung semua *platform* mayor baik itu Windows, Linux/Unix maupun Mac OS, dibangun menggunakan Node JS yang lalu di bingkai menggunakan NWJS [8] yang membuat kode *ter-compile* dan siap di eksekusi.



Gambar 6 Gambar Aplikasi

Selain itu disediakan juga suatu fitur guna *me-manage list keyword & list token* yang akan di gunakan sebagai parameter saat melakukan crawling pada menu proses. Pengujian dilakukan dengan pengujian *black box* dimana penguji cukup dijelaskan mengenai fitur dan batasan pada Twitter, penguji telah mampu memahami alur penggunaan aplikasi dengan sendirinya. Dari pengujian tersebut didapatkan skenario sebagai berikut:

Tabel 1 Pengujian Menu Tokens

Menu Tokens		
Kondisi Pengetesan	Kondisi yang diharapkan	Hasil
1. Isikan semua field 2. Tekan submit	Berhasil masuk	Ok
1. Isikan dengan salah satu isian kosong 2. Tekan submit	Muncul alert atau keterangan	Ok
1. Isikan field 2. Tekan reset	Semua isian kembali ke default	Ok
1. Klik tombol edit	Field isian terisi secara otomatis berdasarkan value yang di edit	Ok
1. Klik tombol hapus	Record terhapus	Ok

Tabel 2 Pengujian Menu Keywords

Menu Keywords		
Kondisi Pengetesan	Kondisi yang diharapkan	Hasil
1. Isikan semua field 2. Tekan submit	Berhasil masuk	Ok
1. Isikan dengan salah satu isian kosong 2. Tekan submit	Muncul alert atau keterangan	Ok
1. Isikan field 2. Tekan reset	Semua isian kembali ke default	Ok
1. Klik tombol edit	Field isian terisi secara otomatis berdasarkan value yang di edit	Ok
1. Klik tombol hapus	Record terhapus	Ok

Tabel 3 Pengujian Menu Proses

Menu Process		
Kondisi Pengetesan	Kondisi yang diharapkan	Hasil
1. Klik "Execute Crawler"	Crawler berjalan dan menghasilkan output sesuai dengancategory	Ok
1. Klik "Stop Crawling"	Crawler berhenti	Ok
1. Klik "execute"	Crawler berjalan hanya untuk category yang dijalankan	Ok

#### 4. KESIMPULAN

Kesimpulan dari penelitian ini adalah:

- 1) Konsep Non-Blocking I/O dapat diterapkan untuk membangun aplikasi yang efektif dan efisien
- 2) Implementasi Non-Blocking I/O bisa diterapkan pada banyak *layer* aplikasi, seperti halnya aplikasi ini, konsep di implementasikan pada saat start aplikasi *clustering*, pada saat proses *crawling* dan pada proses penulisan *output* menjadi *file* menggunakan *Write Stream*.

#### DAFTAR PUSTAKA

- [1] Getting started with Twitter. Available at: <https://support.twitter.com/articles/215585> (Accessed: 7 September 2017).
- [2] NodeJS. Available at: <http://Node.js.org/> (Accessed: 7 September 2017).
- [3] Buettner D., Kunkel J., Ludwig T. (2009). *Recent Advance in Parallel Virtual Machine and Message Passing Interface : Using Non-blocking I/O Operations in High Performance Computing to Reduce Execution Times*. Heidelberg, Germany.
- [4] Fowler, M., Scott, K., Rumbaugh, J., Booch, G., Jacobson, I. and Kobryn, C. (2003). *UML distilled: A brief guide to the standard object modeling language, 3rd edition. 3rd edn*. Boston, MA: Addison-Wesley Educational Publishers.
- [5] Griffin, L., Ryan, K., de Leazar, E. and Botvich, D. (2011). *Scaling instant messaging communication services: A comparison of blocking and non-blocking techniques*. 2011 IEEE Symposium on Computers and Communications (ISCC), . doi: 10.1109/iscc.2011.5983895.
- [6] Guerriero, A., Ragni, F. and Martines, C. (2010). *A dynamic URL assignment method for parallel web crawler*. 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, . doi: 10.1109/cimsa.2010.5611764.
- [7] Jensen, P. (2015) What is NW.js? Available at: <https://dzone.com/articles/what-is-nwjs> (Accessed: 7 September 2016).
- [8] Kaur, H., Talluri, M. and He, J.S. (2015). *Get Twitter information: A collaborative Android application for big data analysis*. 2015 International Conference on Collaboration Technologies and Systems (CTS), . doi: 10.1109/cts.2015.7210475.
- [9] Kim, M.-S. and Gaudiot, J.-L. (2014). *An efficient I/O interface control block design methodology for application-specific MPSoC platforms*. 2014 International Conference on Information Science & Applications (ICISA), . doi: 10.1109/icisa.2014.6847355.
- [10] Mousavi, R. and Gu, B. (2015). *The impact of Twitter adoption on decision making in politics*. 2015 48th Hawaii International Conference on System Sciences, . doi: 10.1109/hicss.2015.576.
- [11] Mubarak bin Naina Hanif, F. and Saptawati, G.A.P. (2014). *Correlation analysis of user influence and sentiment on Twitter data*. 2014 International Conference on Data and Software Engineering (ICODSE), . doi: 10.1109/icodse.2014.7062491.