ISSN 2301-9220

Volume 2 - Nomor 1 - Tahun 2018

JURNAL INFORMATIKA

P R O X I E S

JURNAL ILMIAH NASIONAL
TERBIT BERKALA DUA KALI DALAM SETAHUN

PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS KATOLIK SOEGIJAPRANATA SEMARANG

Proxies	Vol. 2	No. 1	Hal. 1-41	ISSN 2301-9220

PROXIES

JURNAL INFORMATIKA

Alamat Redaksi

Program Studi Teknik Informatika
Gedung Henricus Constant Lt. 8
Unika Soegijapranata
Jl. Pawiyatan Luhur IV/1, Bendan Duwur
Semarang

Daftar Isi

CPU AND GPU PERFORMANCE ANALYSIS ON 2D MATRIX OPERATION

Kwek Benny Kurniawan[1], YB. Dwi Setianto[2]

[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Katholik Soegijapranata

[2]setianto@unika.ac.id

Abstract

GPU or Graphic Processing Unit can be used on many platforms in general GPUs are used for rendering graphics but now GPUs are general purpose parallel processors with support for easily accessible programming interfaces and industry standard languages such as C, Python and Fortran. In this study, the authors will compare CPU and GPU for completing some matrix calculation. To compare between CPU and GPU, the authors have done some testing to observe the use of Processing Unit, memory and computing time to complete matrix calculations by changing matrix sizes and dimensions. The results of tests that have been done shows asynchronous GPU is faster than sequential. Furthermore, thread for GPU needs to be adjusted to achieve efficiency in GPU load.

Keywords: CUDA, GPU, CPU, Parallel

Introduction

GPU or Graphic Processing Unit can be used in many platforms such as smartphones, game consoles, computing and workstations, GPUs are generally used for rendering graphics but now GPUs are general purpose parallel computing platform and programming model that leverages the parallel compute engine in NVIDIA GPUs to solve many complex computational problems in a more efficient way than on a CPU [1].

Compute Unified Device Architecture (CUDA) is a soft- ware platform for massively parallel high-performance computing [1], [2]. By using CUDA, CUDA will facilitate the programmer because it is supported by the use of language c. CUDA is usually used for graphical programming such as digital and video image processing such as image segmentation, image quality change, pattern recognition that can be applied to real-world needs [3]. This project will compare the effect of matrix dimension length to matrix operation and give analysis result used two platform such as CUDA and Java.

Research Method

This project will compare CPU and GPU performance. Author used two processors and one GPU. GPU platform that is used in this project is CUDA, and CPU platform that used in this project is Java.

To compare CPU and GPU Author will try to test GPU and CPU with the different case such as:

1. The effect of matrix dimension with time computation in CPU and GPU. (in CPU will test in different processing unit)

2. The effect of matrix dimension with processing unit usage in CPU and GPU (in % and GPU used 2 Thread to compare [2], [4])
3. The effect of matrix dimension with memory usage in CPU and GPU (in % and GPU used global memory [1], [5], [6])
4. The effect of time on the number of matrix elements in performing different tasks on GPU.

The effect of matrix dimension with time computation, processing unit usage and memory usage in this project tested 4 types of operation matrices with several dimensions of the matrices, and test performance used software monitor GPU and CPU usage memory and time computation in several times and write the analyze compare CPU and GPU used graph.

Results and Analysis

To determine the effect of matrix dimension on computation time, memory usage, use of PU and the influence of matrix length, program testing is matrix, addition and flipping matrix. Testing has been done 3x to get best result and every operation is enhanced with long dimension of 500x500, 1000x1000, 5000x5000 and 10000x10000 tested to 2 processors is dual core E2140 and i5 4460, GPU using 1050Ti.

Below is the results of multiplication effect studies, additions and flipping matrices on different matrix dimensions and explain the results.

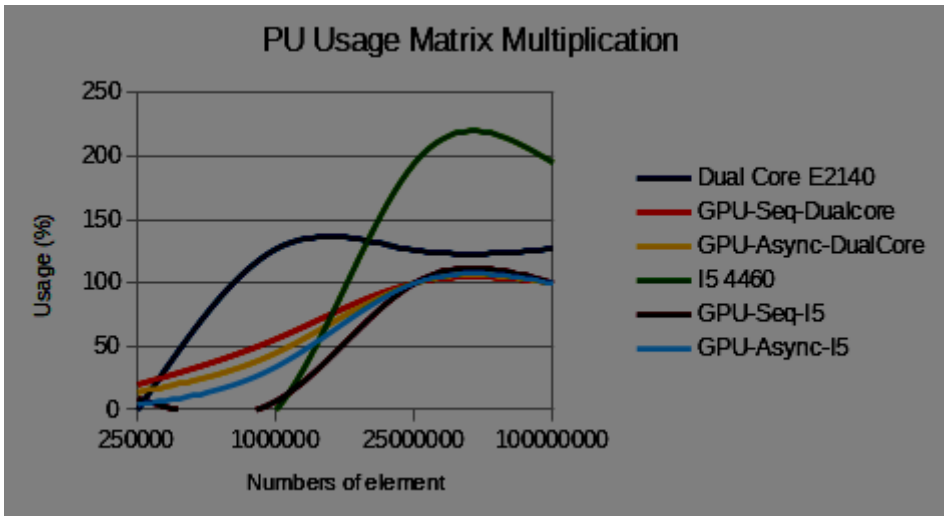


Figure 1: Processing Unit Matrix Multiplication

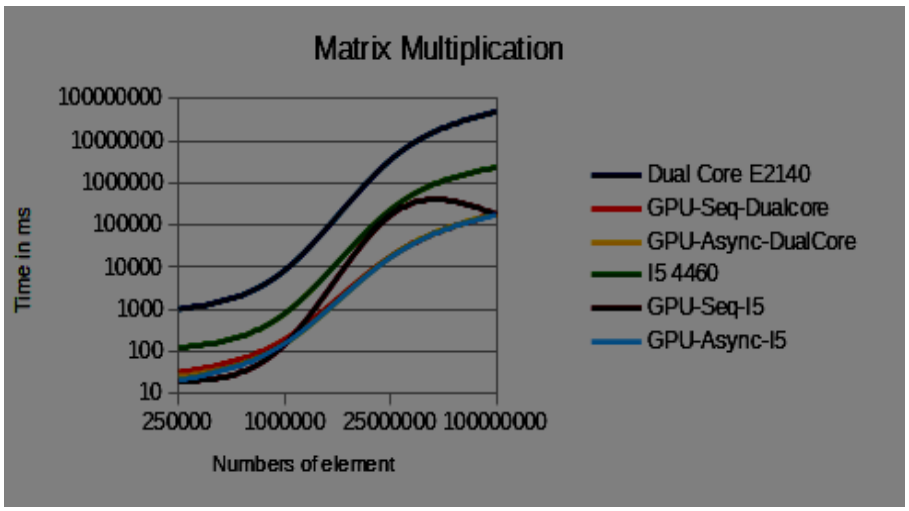


Figure 2: Time Computation Matrix Multiplication

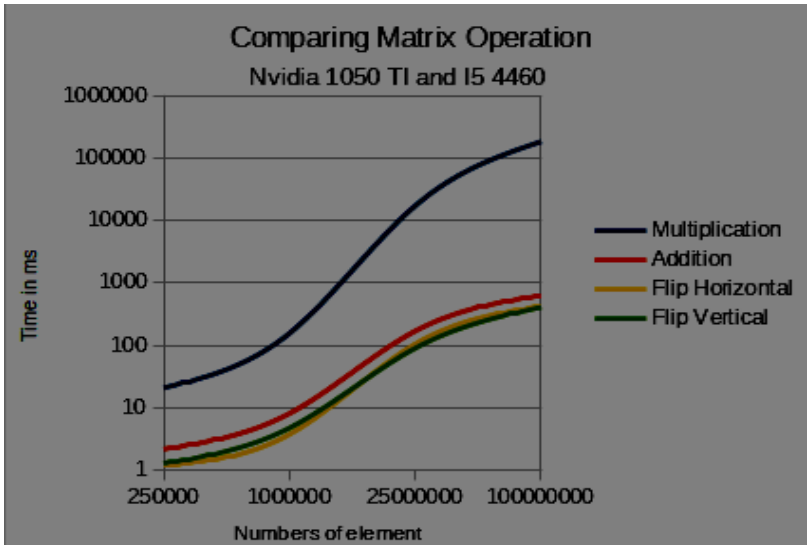


Figure 3: Comparing Matrix Operation

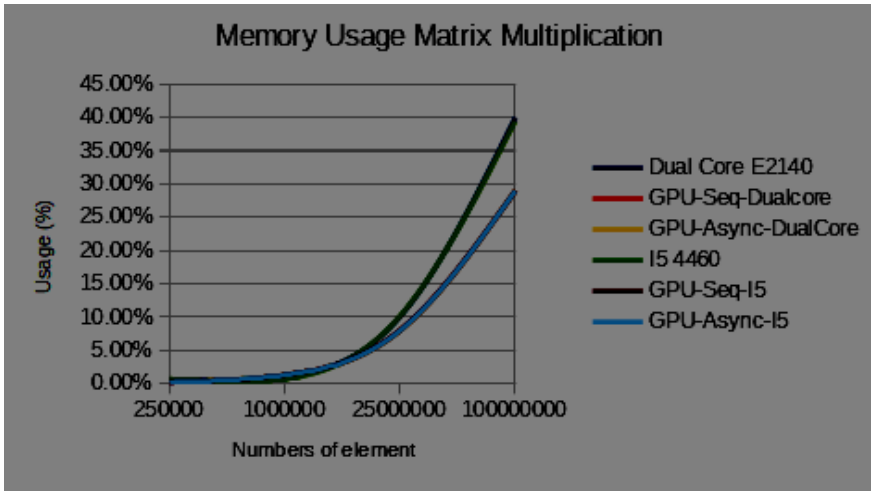


Figure 4: Memory Usage Matrix Multiplication

The graph above shows the multiplication of the computing time matrix on the GPU and CPU has a significant difference. Processor usage tends to be lower than GPU CPU. Memory usage looks quite stable with many spikes and the comparison results show linear results for each given job.

Conclusion

1. The effect of matrix dimension with time computation in CPU and GPU (in CPU will test in different processing unit).

In this work after doing experiment with matrix multiplication, addition and flip (horizontal and vertical) with different matrix size from 500x500, 1000x1000, 5000x5000 and 10000x10000 and monitoring time computation. Time computation showing that GPU matrix multiplication take faster than CPU matrix implementation for data more than 20m above.

2. The effect of matrix dimension with processing unit usage in CPU and GPU (in %). In this work after doing experiment with matrix multiplication , addition and flip (horizontal and vertical) with different matrix size from 500x500, 1000x1000, 5000x5000 and 10000x10000 and monitoring processing unit usage. CPU utilization classified near 200% but in GPU utilization showing 100% but it is can be lower by maximum thread use in because in this project only using 2 Thread to solve.

3. The effect of matrix dimension with memory usage in CPU and GPU (in %). In this work after doing experiment with matrix multiplication , addition and flip (horizontal and vertical) with different matrix size from 500x500, 1000x1000, 5000x5000 and 10000x10000 and monitoring memory usage. Memory to use is directly proportional and look linear.

4. Effect of time on the number of matrix elements in performing different tasks on the GPU. From the test results above shows the same GPU but in use in different PU generate different computational time.

References

[1] NVIDIA, “CUDA C Programming Guide,” no. December, 2016.

[2] T. R. Halfhill, “Parallel Processing with CUDA,” Microprocessor Report, pp. 1–8, 2008.

[3] B. Kurniawan, T. B. Adji, and N. A. Setiawan, “Analisis Perbandingan Komputasi GPU dengan CUDA dan Komputasi CPU untuk Image dan Video Processing,” Seminar Nasional Aplikasi Teknologi Informasi (SNATI), vol. 1, no. 1, pp. 25–31, 2015.

[4] V. Volkov, “Better performance at lower occupancy,” Proceedings of the GPU Technology Conference, pp. 1–75, 2010.

[5] Khoirudin and J. Shun-Liang, “GPU application in CUDA memory,” Advanced Computing: An International Journal, vol. 6, no. 2, pp. 1–10, 2015, doi: 10.5121/acij.2015.6201.

[6] N. Corporation, “NVIDIA CUDA Architecture,” Compute, no. April, 2009.NVIDIA, “CUDA C Programming Guide,” no. December, 2016.

WORD SEARCH USING BOYER-MOORE ALGORITHM

Prana Pangestu[1], Shinta Estri Wahyuningrum[2]
[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata
[1]pranapangestu@gmail.com, [2]shinta@unika.ac.id

Abstract

Boyer-Moore is one of the algorithms used for the search words that is said to be one of the fastest to complete the search process. This project aims to implement hash table and see the effect it has in the searching process. A hash table is a table that contains key that maps to a value using a hash function. By doing so, the hash table should improve the speed of searching process. The vanilla version of Boyer-Moore will be used as the baseline to test the speed and accuracy of this method and check whether further improvement are needed.

Keywords: *boyer-moore, hash table, string searching, searching, sequential search algorithm*

Pendahuluan

Pencarian kata sering dilakukan untuk mencari kata yang dicari didalam sebuah file atau dokumen. Penelitian untuk pencarian kata perlu dilakukan agar bisa memperoleh sebuah cara yang efisien untuk mempercepat proses pencarian dan ketepatan hasil.

Ada berbagai macam algoritma untuk pencarian kata yang sering digunakan seperti brute force dan linear search. Kelebihan dari brute force adalah mudah dimengerti, bisa digunakan untuk berbagai macam masalah yang lain tetapi algoritma ini mempunyai kelemahan tidak bisa digunakan untuk permasalahan yang membutuhkan penyelesaian cepat dan cenderung lambat. Linear Search juga sering digunakan dalam proses pencarian kelebihan algoritma ini relatif cepat jika data tidak terlalu kompleks dan mudah digunakan. Kelemahan nya adalah jika data terlalu banyak/kompleks maka pencarian akan berlangsung lambat.

Boyer-Moore adalah algoritma yang dianggap efisien dalam mencari kata/string karena membandingkan karakter dari sebelah kanan sehingga mempercepat proses pencarian. Jika tidak sama maka langsung bergeser ke karakter selanjutnya.

Landasan Teori

Pada jurnal Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android. Kencana Wulan Arganingrum & Seng Hansun mengimplementasikan algoritma boyer-moore pada kamus kedokteran berbasis android [1]. Boyer-Moore digunakan untuk mencari kata yang telah disimpan sebelum nya di database. Hasil akhir dari penelitian ini adalah aplikasi android kamus kedokteran.

Jurnal kedua berjudul Implementasi Algoritma Pencarian Sequential Search pada Ensiklopedia Ikan Hias Air Tawar Berbasis Android. Haerul Umam

mengimplementasikan algoritma pencarian sequential search pada data jenis ikan hias air tawar [2]. Hasil akhir dari penelitian ini adalah aplikasi android untuk ensiklopedia.

Jurnal ketiga adalah On Improving the average case of Boyer-Moore String Matching Algorithm. Zhu Rui Feng and Tadao Takaoka menjelaskan bagaimana algoritma boyer-moore bisa digunakan untuk pencarian string [3]. Jurnal ini juga menjelaskan aturan yang berlaku di algoritma boyer-moore dan cara kerja algoritma boyer-moore.

Pada e-book berjudul Algorithm (Fourth Edition). Robert Sedgewick and Kevin Wayne menjelaskan tentang implementasi hash table. E-book ini menjelaskan struktur data Hash Table, cara kerja dari Hash Table, bentuk dari Hash Table dan implementasi dalam bahasa pemrograman [4].

Jurnal keempat berjudul Approximate Multiple String Search. Robert Muth and Udi Manber membahas tentang hash table yang bisa digunakan untuk pencarian string [5].

R. S, Boyer & J. S, Moore pada jurnal The Boyer-Moore Theorem Prover and Its Interactive Enhancement menjelaskan tentang penemuan algoritma boyer-moore, cara kerja boyer-moore dan kegunaan nya [6].

Metodologi Penelitian

1. Studi Literatur

   Mencari jurnal yang berkaitan dengan string match, boyer-moore dan hash table. Mempelajari aturan dalam algoritma boyer-moore dan cara kerja boyer-moore.

2. Membuat Desain Program

   Langkah selanjutnya membuat desainn dan flowchart program. Merancang desain program untuk Pre-Processing dari input user, membuat BadChar Table dari input kata yang dicari, dan membuat proses pencarian menggunakan Boyer-Moore.

3. Implementasi

   Membuat program menggunakan bahasa pemrograman PHP yang digunakan untuk implementasi algoritma boyer-moore, membuat hash table, membuka dan membaca text dari input user, upload text file dan pencarian kata.

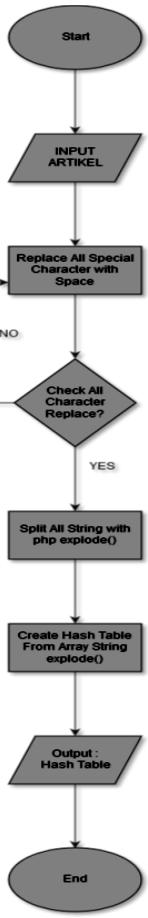
4. Testing:

   a. Membandingkan boyer-moore hash table dengan boyer-moore dan membandingkan boyer-moore hash table dengan sequential search.

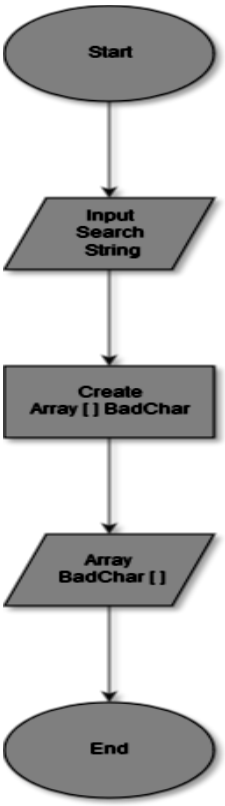
   b. Menganalisis hasil pencarian dan waktu eksekusi program.

   c. Mencari kata dengan algoritma Boyer-Moore Hash Table.

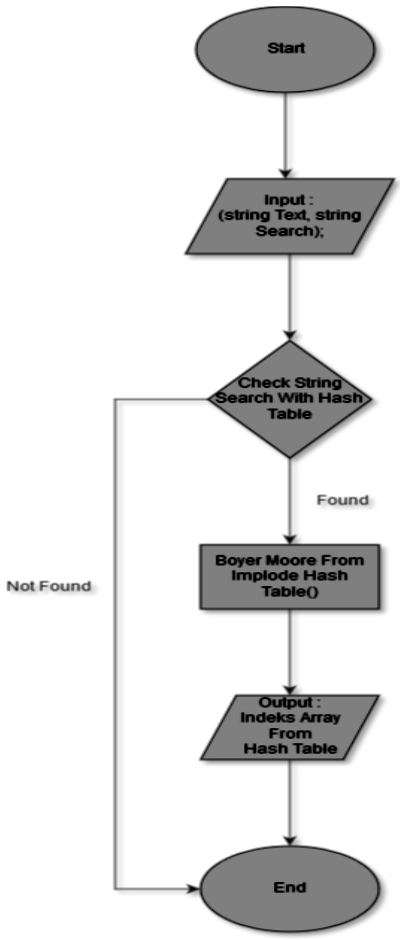
Hasil dan Pembahasan



Gambar 1: Flowchart pre-order



Gambar 2: Bad Char Table



Gambar 3: Final Process

Pada Gambar.1 menggambarkan flowchart pre-order proses sebelum menjalankan process algoritma Boyer-Moore. Pertama input file dengan format .txt kemudian semua spesial karakter di tiap string akan diganti dengan spasi. Setelah proses selesai maka dibuat hash table dari string yang telah diolah.

Setelah melakukan pre-order proses maka langkah selanjutnya membuat bad Char tabel yang digunakan untuk menghitung pergeseran pada Boyer-Moore seperti pada Gambar 2.

BadChar value = panjang dari string yang dicari – index -1

Contoh: TOOTH

T O O T H

0 1 2 3 4

Pilih semua huruf yang berbeda = TOH

Panjang string = 5

Tabel 1: Bad Char Table

Letter	**T**	**O**	**H**	*****
Value	1	2	5	5

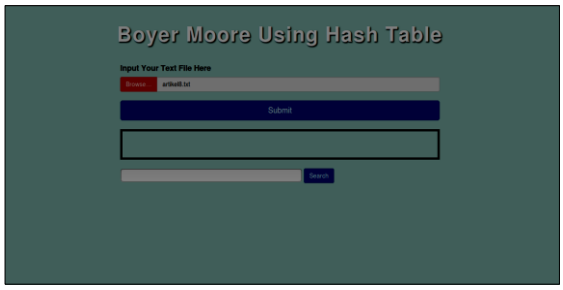
T = 5 – 3 – 1, O = 5 – 2 – 1, H = 5 (Huruf terakhir = panjang string, jika belum didefinisikan)

* = panjang string.

Setelah Pre-Process dan BadChar sudah dilakukan maka dilanjutkan dengan proses Boyer-Moore Gambar.3. Boyer-moore dilakukan dengan memanggil function boyer-moore yang memiliki dua parameter yang berisi teks dan kata yang akan dicari. Kemudian dari kata yang dicari akan dihitung jumlah ascii dari tiap karakter nya kemudian di modulo dengan 26 untuk mendapatkan nomor indeks yang baru. Jika nomor indeks nya tidak terdapat dalam hash table yang telah dibuat maka proses akan dihentikan. Apabila nomor indeks terdaftar maka baru dilakukan proses Boyer-Moore.

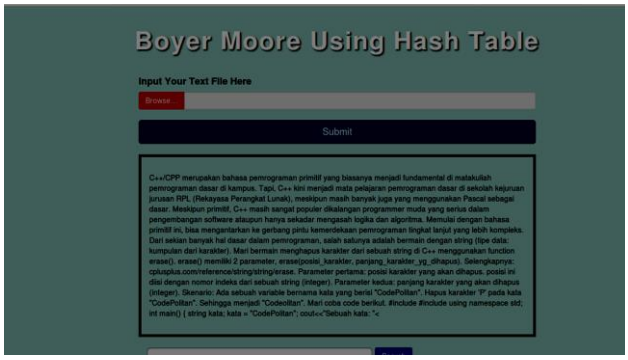
Pada program tersebut terdapat form untuk upload file, tombol submit untuk menampilkan file yang telah diupload, display teks dan form pencarian kata.

1. Menampilkan nama file yang akan diupload.



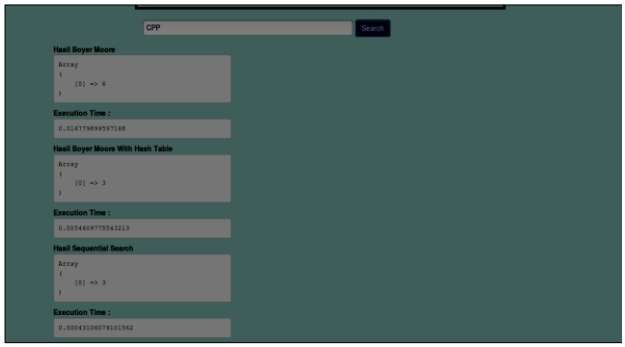
Gambar 4: upload file

2. Menampilkan hasil dari file yang telah diupload.



Gambar 5: Hasil dari teks yang telah diupload

3. Mencari kata dan menampilkan hasil perbandingan serta waktu eksekusi program dari Boyer-Moore, Boyer-Moore dengan Hash Table dan Sequential Search.



Gambar 6: Hash Table dan Sequential Search

Tabel 2: Boyer-Moore

Article	String Length	String Search	Result	Execution Time
artikel8.txt	2516	CPP	Find	0
artikel9.txt	4001	Sublime	Find	0
renungan.txt	2112	Kasih	Not find	0
renungan2.txt	1976	Paulus	find	0
artikel15.txt	5764	PHP	Not Find	0

Tabel diatas menunjukan hasil dari percobaan menggunakan boyer-moore.

Tabel 3: Result of Boyer-Moore Hash Table

Article	String Length	String Search	Result	Execution Time
artikel8.txt	2516	CPP	Find	0.01
artikel9.txt	4001	Sublime	Find	0.01
renungan.txt	2112	Kasih	Not find	0
renungan2.txt	1976	Paulus	find	0
artikel15.txt	5764	PHP	Not Find	0.01

Tabel diatas menunjukan hasil dari percobaan menggunakan boyer-moore dengan hash table.

Tabel 4: Result of Sequential Search

Article	String Length	String Search	Result	Execution Time
artikel8.txt	2516	CPP	Find	0
artikel9.txt	4001	Sublime	Find	0
renungan.txt	2112	Kasih	Not find	0
renungan2.txt	1976	Paulus	find	0
artikel15.txt	5764	PHP	Not Find	0

Tabel diatas menunjukan hasil dari percobaan menggunakan sequential search.

Dari ketiga tabel tersebut menunjukan bahwa pencarian dengan menggunakan boyer-moore hash table tidak lebih cepat dari boyer-moore dan sequential search. Panjang teks dan kata yang akan dicari juga berpengaruh terhadap waktu eksekusi program.

Kesimpulan

Kesimpulan dari project ini Boyer-Moore Hash Table tidak lebih cepat dari Boyer-Moore karena pada Boyer-Moore Hash Table text file harus dibuat menjadi hash table terlebih dahulu. Sehingga membutuhkan proses yang lebih lama. Tapi program ini lebih mudah digunakan karena menghasilkan no indeks dari kata yang dicari.

Daftar Pustaka

[1] K. Argakusumah and S. Hansun, "Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android," *ULTIMATICS*, vol. 6, p. 70, Dec. 2014, doi: 10.31937/ti.v6i2.340.

[2] H. Umam, S. Hardienata, and A. Chairunnas, "Implementasi Algoritma Pencarian Sequential Search pada Ensiklopedia Ikan Hias Air Tawar berbasis Android."

[3] F. Rui and TakaokaT, "On improving the average case of the Boyer-Moore string matching algorithm," *Journal of Information Processing*, Jul. 1988, Accessed: Feb. 04, 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/50803.50808.

[4] R. Sedgewick and K. Wayne, *Algorithms, 4th Edition*. Addison-Wesley, 2011.

[5] R. Muth and U. Manber, "Approximate multiple string search," in *Combinatorial Pattern Matching*, Jun. 1996, pp. 75–86, doi: 10.1007/3-540-61258-0_7.

[6] R. S. Boyer, M. Kaufmann, and J. S. Moore, "The Boyer-Moore theorem prover and its interactive enhancement," *Computers & Mathematics with Applications*, vol. 29, no. 2, pp. 27–62, Jan. 1995, doi: 10.1016/0898-1221(94)00215-7.

PENJADWALAN SATPAM JAGA DENGAN MENGGUNAKAN ALGORITMA GENETIKA

Ronny Loekito[1], Hironimus Leong[2]

[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata

[1]ronnyloekito@gmail.com, [2]marlon.leong@unika.ac.id

Abstract

Scheduling is to manage life easier and more effective. It informs us about what to do, and when we do it. Many scheduling are still arranged manually, including security scheduling. This project is to generate security schedule automatically. With genetic algorithm that usually used in schedule optimization and Java programming language. The expectation of this project is to make security scheduling easier and more effective without any collisions.

Keywords: *Genetic Algorithm, Security Scheduling, Optimization*

Pendahuluan

Jaman sekarang ini, penjadwalan itu penting untuk mengatur kegiatan sehari-hari lebih mudah dan efektif. Dengan jadwal, kita jadi tahu apa yang harus dilakukan dan kapan kita melakukannya. Salah satu penjadwalan dalam kehidupan adalah penjadwalan satpam jaga. Dalam penjadwalan satpam, dibutuhkan data mengenai informasi satpam (nama dan jenis kelamin), jam kerja, dan pos jaga. Untuk mengoptimasi penjadwalan, dapat diselesaikan dengan banyak cara, salah satunya adalah dengan algoritma genetika.

Algoritma genetika diawali dengan inisialisasi dari data-data yang ada yang membentuk individu. Lalu dilanjutkan dengan penghitungan nilai fitness tiap individu. Setelah itu diseleksi untuk memilih individu dengan fitness terbaik. Lalu dilanjutkan dengan crossover dan mutasi. Crossover itu proses dimana memilih antara 2 individu hasil seleksi lalu saling disilangkan data antar masing-masing individu. Sedangkan mutasi adalah proses memilih suatu individu untuk dimutasi datanya. Setelah itu di evaluasi nilai fitness lagi, dan akhirnya muncul hasil jadwal akhir.

Landasan Teori

Literatur dengan judul "Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika" oleh Nia Kurnia Mawaddah (2006) menjadi salah satu literatur dalam proyek ini. Jadwal dalam literatur ini berisi dosen, waktu ujian, dan juga ruangan yang digunakan. Inputan data yang diperlukan dalan proses penjadwalan ini adalah mata kuliah, dosen, ruang kelas, dan waktu. Batasan dalam literatur ini adalah dosen jaga harus datang saat ujian dilakukan.

Menurut Achmad Hidayatno (2010) dalam literatur berjudul "Penerapan Algoritma Genetika Pada Perencanaan Lintasan Kendaraan", proyek ini menggunakan algoritma genetika dengan menghitung dan memperoleh jalur pada permukaan yang telah dipisah

menjadi bagian-bagian kecil. Pada permukaan yang ada halangan misalnya dataran tinggi atau ada sesuatu yang harus dihindari. Tujuan dari proyek ini untuk mencari jalur terpendek dari satu lokasi ke lainnya.

Menurut Mery Hanita (2011) pada literatur berjudul "Penerapan Algoritma Genetika Pada Penjadwalan Mata Kuliah". Penjadwalan ini berisi mata kuliah, dosen, ruang kelas, waktu, dan mahasiswa yang mengambil mata kuliah itu. Itu juga tergantung pada kemungkinan dosen tidak bisa mengajar karena ada aktivitas akademik lain atau dosen mengajar lebih dari satu mata pelajaran di hari dan jam yang sama, mahasiswa yang mengambil mata kuliah yang bertabrakan, dan ketersediaan ruang kelas apakah sedang dipakai atau tidak. Hasil penjadwalan akan ditampilkan di GUI.

Menurut Aulia Fitrah (2006) pada jurnal berjudul "Penerapan Algoritma Genetika Pada Persoalan Pedagang Keliling (TSP)". Tujuan dari proyek ini adalah untuk memperoleh jalur yang termurah untuk mengunjungi seluruh kota, kunjungan tiap kota hanya boleh sekali, dan lalu kembali ke kota awal. Biaya perjalanan dihitung berdasarkan jarak, waktu, dan bahan bakar.

Pada "Optimasi Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika" oleh Zaini, Hidayat, dan Regasari (2014), literatur ini tentang bagaimana untuk menyusun jadwal jika jumlah ruang kelas hanya sedikit dan jumlah dosen juga terbatas. Proyek ini menggunakan algoritma genetika untuk menyusun jadwal menggunakan berbasis web. Parameter proyek ini adalah jumlah pelajaran dalam seminggu, guru, ruang kelas, hari, dan total jam kegiatan mengajar.

Pada literatur "Produce Cross Word Answer Using Genetic Algorithm" oleh Donny Budiarto (2014), tujuan proyek ini adalah menghasilkan kemungkinan jawaban pada teka teki silang berdasarkan huruf yang ditemukan oleh pengguna. Jawaban tidak hanya dalam bahasa Indonesia, tetapi juga bahasa Inggris. Parameter di proyek ini menggunakan inputan pengguna seperti jumlah huruf dan huruf apa saja untuk menyusun kata. Hasil proyek ini akan menampilkan kemungkinan jawaban berdasarkan parameter.

Pada literatur "Algoritma Genetika Dalam Pemilihan Spesifikasi Komputer" oleh Anthony (2013), tujuan proyek ini untuk membantu pengguna untuk mengoptimalkan dalam pemilihan spesifikasi komputer yang sesuai kebutuhan. Parameter proyek ini adalah tujuan dari penggunaan komputer apakah untuk bermain game atau menonton film atau kebutuhan kantor.

Pada "Penerapan Algoritma Genetik Pada Permainan Catur Jawa" oleh Nico Saputro dan Erdo Dirgagautama (2003), pada catur jawa ada tujuh strategi berdasarakan situasi papan catur untuk memilih jalan yang dilalui komputer dan tiap strategi memiliki bobot masing-masing. Algoritma genetika digunakan untuk mencari bobot pada tiap strategi. Parameter dari proyek ini adalah inputan pengguna dalam permainan.

Metodologi Penelitian

1. Analisis

Langkah awal dari proyek ini adalah dengan menganalisa data apa yang diperlukan di dalam penjadwalan satpam, antara lain data satpam, pos jaga, dan jam kerja. Tujuannya adalah membuat jadwal dengan menggunakan algoritma genetika.

2. Membuat Desain Program

Langkah selanjutnya adalah membuat desain program berdasarkan data dari langkah pertama. Desain program terdiri dari input user, lalu input itu akan diproses dengan algoritma genetika, dan hasil output akan ditampilkan di GUI. Data yang diperoleh dari input user antara lain nama dan jenis kelamin satpam, jumlah pos jaga, mulai jam kerja, serta waktu tiap shift kerja. Dari data tersebut algoritma genetika akan menginisialisasi sebuah populasi yang terdiri dari beberapa individu. Dan nilai fitness masing-masing individu akan dihitung. Lalu semua individu itu akan diseleksi dengan menggunakan teknik seleksi *roulette wheel*. Seleksi berfungsi untuk memilih individu-individu terbaik. Lalu dilanjutkan dengan crossover, dimana proses ini akan menggabungkan data antara suatu individu dengan individu lain. Selanjutnya proses mutasi yang akan mengacak isi dari suatu individu yang terpilih. Setelah semua proses itu, maka akan terbentuk populasi baru, dan populasi itu akan di evaluasi juga nilai fitness tiap individunya. Setelah itu baru akan terpilih lah individu terbaik untuk ditampilkan di GUI.

3. Implementasi

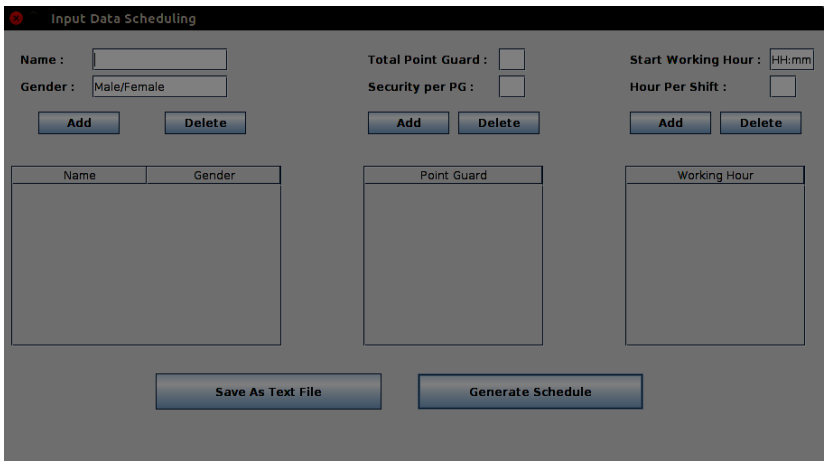
Setelah semua langkah di atas, program siap untuk dibuat dengan menggunakan bahasa pemrograman Java dan algoritma genetika. Langkah ini termasuk pembuatan coding dan menampilkan hasil program di GUI. Setelah selesai membuat program, maka perlu dilakukan pengetesan. Akan dicoba menjalankan program untuk memperoleh hasil dan mengecek apakah ada kesalahan pada hasilnya.

4. Membuat Laporan

Langkah terakhir adalah pembuatan laporan proyek. Laporan ini berisi data apa saja dari inputan user melalui GUI, proses untuk mencari data optimal, sampai hasil yang ditampilkan pada GUI. Hasil dari program ini adalah jadwal yang terdiri dari hari dan jam kerja, nama dan jenis kelamin satpam, serta pos jaga tiap satpam.

Hasil dan Pembahasan

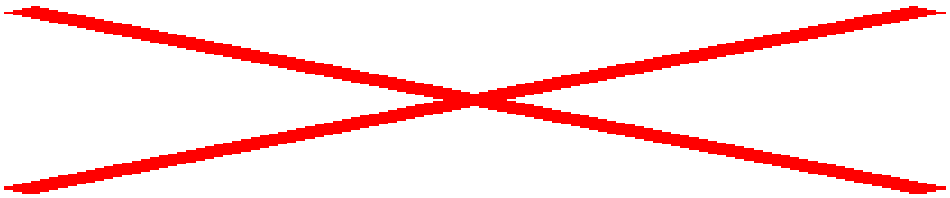
Pogram ini membutuhkan inputan user seperti data satpam, jumlah pos jaga, dan jam mulai jaga serta jam tiap shift nya.



Gambar 1: Input User

Setelah itu lalu klik save as text file untuk menyimpan data dalam file .txt. Lalu klik generate schedule untuk mendapatkan jadwal di GUI.

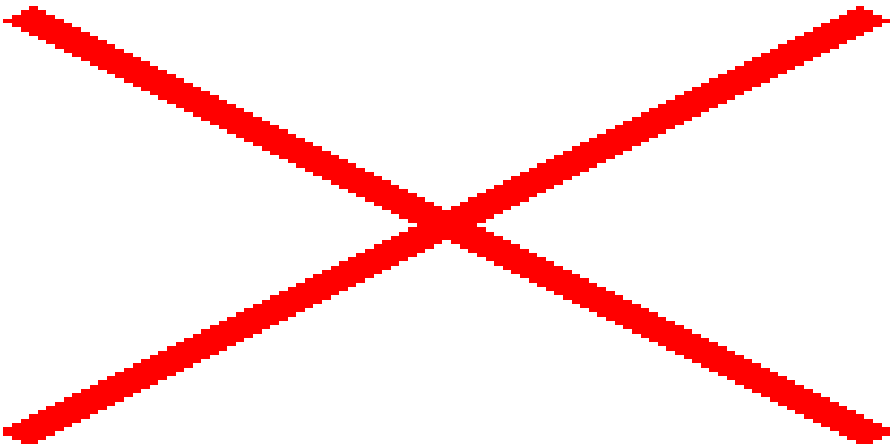
Proses awal dari algoritma genetika ini adalah generate data dari user input tadi secara acak. Misalnya Arif | Male | P1 | 06:00 (nama | gender | pos jaga | shift jaga). Dan tersimpan dalam array 2 dimensi. Acakan data terbentuk sebanyak 7 (total hari dalam seminggu) x jumlah satpam jaga dalam sehari dan membentuk satu individu. Dan generate itu akan berulang sampai membentuk 10 individu. 10 individu itu akan tergabung menjadi 1 populasi.



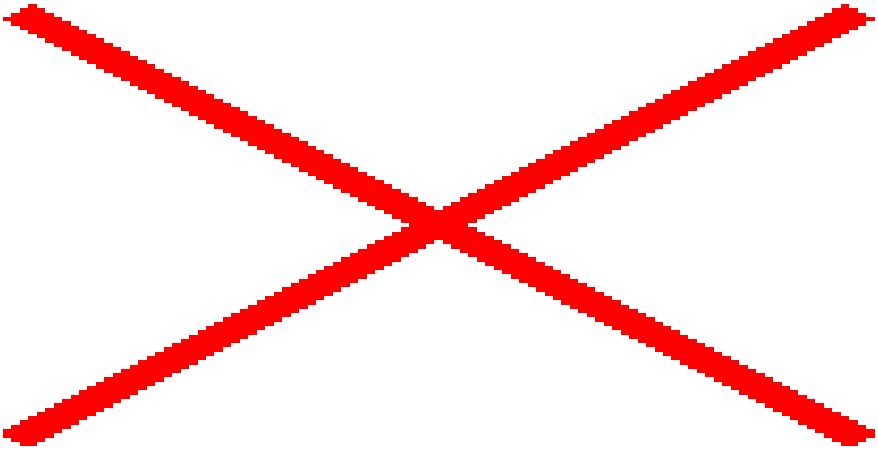
Gambar 2: Contoh Satu Individu

Setelah itu tiap individu akan di evaluasi nilai fitness nya, dengan adanya aturan maksimal satpam dalam tiap pos yang sesuai inputan user dan satpam wanita tidak boleh jaga di atas jam 18:00. Jika data dari array tersebut melanggar aturan itu maka akan mendapat poin 0, sedangkan data yang tidak melanggar akan mendapat poin 1.

Lalu ada proses seleksi, pada program ini menggunakan seleksi roulette wheel. Di mana seleksi ini akan memilih beberapa individu untuk diproses lebih lanjut. Proses selanjutnya adalah crossover satu titik, di mana proses ini akan memilih dua individu dan saling bertukar datanya.

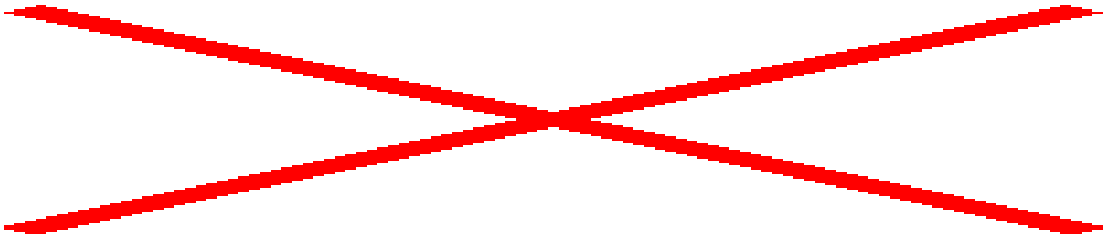


Gambar 3: Sebelum Crossover

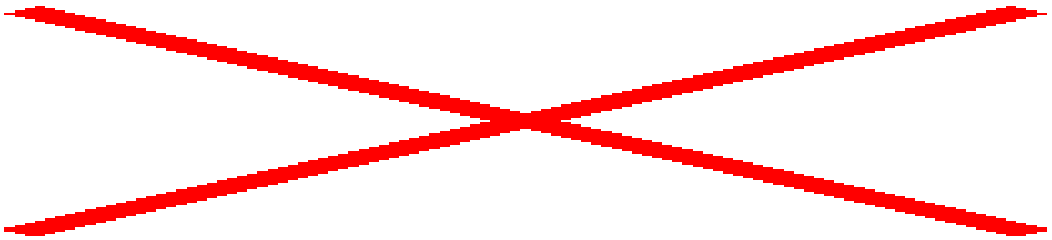


Gambar 4: Sesudah Crossover

Proses berikutnya adalah mutasi, probabilitas terjadinya mutasi ini adalah 0,03. Jadi kemungkinan terjadinya mutasi ini sangatlah kecil. Proses ini akan memilih satu individu untuk dimutasi.

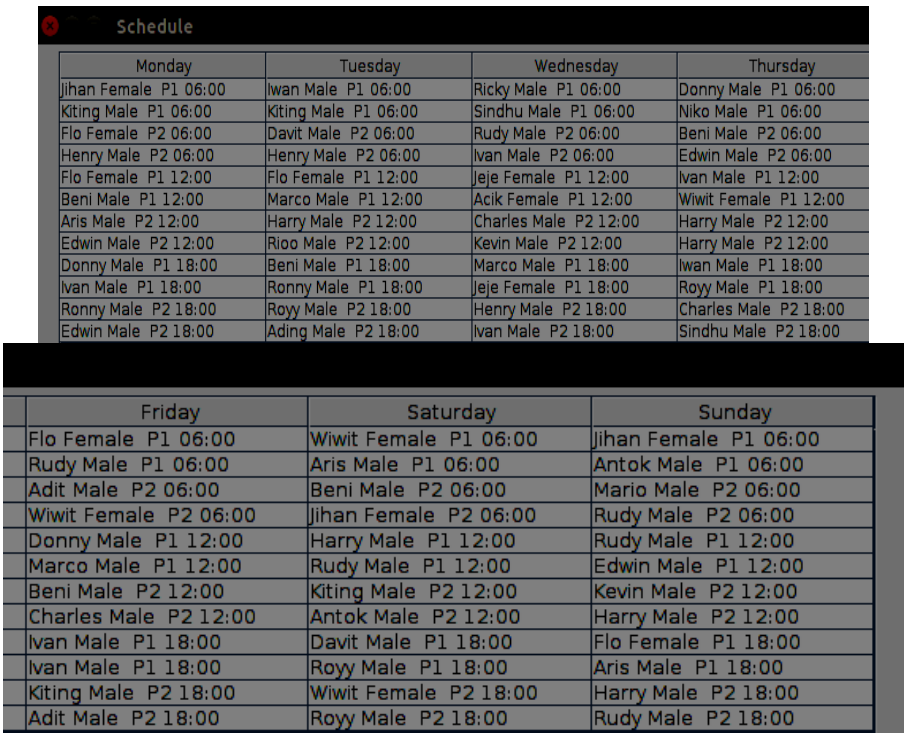


Gambar 5: Sebelum Mutasi



Gambar 6: Sesudah Mutasi

Lalu proses terakhir adalah evaluasi kembali nilai dari tiap individu. Individu dengan nilai terbaik akan ditampilkan pada GUI sebagai jadwal akhir.



Gambar 7: Jadwal Pada GUI

Kesimpulan

Algoritma genetika dapat diaplikasikan pada penjadwalan yang lebih mudan dan menghemat waktu daripada membuat jadwal secara manual. Algoritma genetika juga dapat membuat jadwal tanpa adanya tabrakan jadwal karena di dalam genetika ada proses evaluasi yang berfungsi untuk menilai jadwal, jika jadwal itu ada yang tabrakan maka akan mengulangi proses genetika lagi.

Daftar Pustaka

[1] Anthony, “Algoritma Genetika Dalam Pemilihan Spesifikasi Komputer”, Universitas Sumatra Utara, 2013

[2] Donny Budiyanto, “Produce Cross Word Answer Using Genetic Algorithm”, Universitas Katolik Soegijapranata, 2014

[3] Aulia Fitrah, Achmad Zaky, dan Fitrasani, “Penerapan Algoritma Genetika Pada Persoalan Pedagang Keliling (TSP)”, Institut Teknologi Bandung, 2006.

[4] Mery Hanita, “Penerapan Algoritma Genetika Pada Penjadwalan Mata Kuliah”, Universitas Bengkulu, 2011.

[5] Achmad Hidayatno, Darjat, dan Hendry HLT, “Penerapan Algoritma Genetika Pada Perencanaan Lintasan Kendaraan”, Universitas Diponegoro, 2010.

[6] Andhika Lady Maharsi, “Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika”, Universitas Negeri Yogyakarta, 2013.

[7] Nia Kurnia Mawaddah dan Wayan Firdaus Mahmudy, “Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika”, Universitas Brawijaya, 2006.

[8] Nico Saputro dan Erdo Dirgagautama, “Penerapan Algoritma Genetik Pada Permainan Catur Jawa”, Universitas Katolik Parahyangan, 2003.

ANALISA KEMUNGKINAN ALGORITMA SHA256 & ALGORITMA SCRYPT DALAM MENEMUKAN BLOK BARU PADA TEKNOLOGI BLOCKCHAIN

Novan Ageng Mulyadi[1], YB. Dwi Setianto[2]

[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata

[1]14k10074@student.unika.ac.id, [2]setianto@unika.ac.id

Abstract

In cryptocurrency, a block is a record of new transactions that could mean the location of cryptocurrency, medical data, or even voting records. Once each block is completed it's added to the chain, creating a chain of blocks: a blockchain. Because cryptocurrencies are encrypted, processing any transactions means solving complicated math problems using specified algorithm like SHA256 and Scrypt. People who solve these equations are rewarded with cryptocurrency in a process called mining. In this study the authors will find out the comparison of SHA256 and Scrypt to work on the blockchain. To find out the comparison between SHA256 and Scrypt some test will be done in this research. The test were given to both algorithm on some test System to observe which algorithm that has highest probability in finding new block. the results of the tests done in this paper show that Scrypt is much slower to hash than SHA256 but the Scrypt algorithm has greater chance to find a new block.

Keywords: *SHA256, Scrypt, Proof-of-Work, Blockchain, Comparison*

Pendahuluan

Penelitian ini membandingkan probabilitas untuk menemukan blok baru antara Algoritma SHA256 dan Algoritma Scrypt yang terdiri dari enam bab. Bab pertama adalah tentang latar belakang yang membandingkan SHA256 dan Scrypt secara umum, bab kedua berisi tentang referensi penelitian seperti proses publikasi elektronik dan dokumentasi, bab ketiga akan menjelaskan Metodologi secara umum seperti platform yang digunakan, eksperimen yang akan diuji dan bagian yang diamati untuk mengumpulkan data, bab keempat akan menggambarkan sistem pengujian dan program yang akan digunakan untuk mengamati, di bab kelima berisi hasil eksperimen yang telah dilakukan dan bab keenam terakhir berisi hasil kesimpulan dari penelitian ini.

Landasan Teori

Menurut Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, Robbert van Renesse, Bitcoin adalah mata uang kripto terdesentralisasi terdesentralisasi yang secara implisit mendefinisikan dan menerapkan konsensus [1]. Bitcoin menggunakan protokol blockchain untuk membuat serial transaksi mata uang Bitcoin di antara penggunanya. Mesin negara direplikasi mempertahankan saldo dari pengguna yang berbeda, dan transisinya adalah transaksi yang memindahkan dana di antara mereka. Mesin negara ini dikelola oleh node-node sistem, yang disebut penambang.

Menurut Roman Beck, Jacob Stenum Czepluch, Nikolaj Lollike, Simon Malone, Blok A berisi data semua transaksi dalam jangka waktu tertentu, dan referensi ke blok sebelumnya [2]. Kriptografi yang digunakan untuk membuat blok berbeda tergantung pada protokol blockchain yang digunakan, tetapi pada dasarnya kita dapat melintasi seluruh blockchain dan menemukan setiap transaksi yang pernah dilakukan, semua jalan kembali ke blok pertama, yang disebut blok genesis. Algoritma Hashing digunakan untuk memastikan bahwa semua blok terbentuk dengan baik dan tidak dirusak, dan dengan demikian blockchain membuat dirinya aman dan hampir tidak bisa dipecahkan. Blockchain tidak disediakan dari satu server, tetapi dijalankan pada jaringan komputer yang tersebar luas sebagai buku besar yang didistribusikan. Semua peserta jaringan menyimpan semua data di blockchain, dan semuanya bekerja sama dalam mengembangkannya. Komputer-komputer ini sering disebut penambang. Tergantung pada protokol blockchain, ini akan bersaing untuk membentuk blok baru yang kemudian ditambahkan ke blockchain ketika dipilih melalui skema konsensus.

Menurut Arthur Gervaisy, Ghassan O. Karamez, Damian Gruber, Srdjan Capkun, ketentuan privasi karena integrasi filter Bloom di klien SPV [3]. Tunjukkan bahwa filter Bloom menyebabkan kebocoran privasi yang serius dalam implementasi klien SPV yang ada. Lebih khusus lagi, bahwa sejumlah besar alamat pengguna klien SPV yang memiliki sejumlah kecil alamat Bitcoin (misalnya, <20) bocor oleh satu filter Bloom. Selain itu, sejumlah besar alamat pengguna bocor jika musuh dapat mengumpulkan dua filter Bloom berbeda yang dikeluarkan oleh node yang sama, terlepas dari target false positive rate dari filter, dan jumlah alamat yang dimiliki oleh pengguna. Mengingat kebocoran informasi seperti itu dapat sangat merusak privasi pengguna.

Menurut Arthur Gervaisy, Hubert Ritzdorfy, Ghassan O. Karamez and Srdjan Capkun, perbedaan antara jumlah input dan output dari suatu transaksi dikumpulkan dalam bentuk biaya oleh penambang Bitcoin [4]. Penambang adalah rekan, yang berpartisipasi dalam pembuatan blok Bitcoin. Blok-blok ini dihasilkan dengan menyelesaikan skema proof-of-work (PoW) berbasiskan hash, para penambang harus menemukan nilai nonce yang, ketika di-hash dengan field tambahan, hasilnya di bawah nilai target yang diberikan. Jika seperti itu ditemukan, penambang kemudian memasukkannya ke dalam blok baru sehingga memungkinkan setiap entitas untuk memverifikasi PoW. Karena setiap blok terhubung ke blok yang dihasilkan sebelumnya, blockchain Bitcoin tumbuh pada generasi blok baru di jaringan.

Menurut Arthur Gervais, Vasileios Glykantzis, Ghassan O. Karame, Hubert Ritzdorf, Karl Wüst, Srdjan Capkun, mekanisme konsensus bukti kerja (PoW) adalah mekanisme konsensus terluas yang terluas dalam blockchain yang ada [4]. PoW diperkenalkan oleh Bitcoin dan mengasumsikan bahwa setiap rekan menilainya dengan "kekuatan komputasi" nya dengan memecahkan bukti contoh kerja dan membangun blok yang sesuai. Bitcoin, misalnya, menggunakan PoW berbasis hash yang memerlukan pencarian nilai nonce, sehingga ketika di-hash dengan parameter blok tambahan, nilai hash harus lebih kecil dari nilai target saat ini. Ketika seperti itu ditemukan, penambang menciptakan blok dan meneruskannya pada lapisan jaringan.

Menurut Jo¨el Alwen, Binyi Chen, Krzysztof Pietrzak, Leonid Reyzin, Stefano Tessaro, Scrypt adalah calon MHF sederhana yang dirancang oleh Percival, dan dijelaskan dalam RFC 7914 [5]. Telah digunakan dalam sejumlah cryptocurrency dan telah menjadi inspirasi bagi Argon2d, salah satu pemenang dari kata sandi baru-baru ini kompetisi-hashing. Meskipun popularitasnya, tidak ada batas bawah yang ketat pada kompleksitas memorinya yang diketahui.

Menurut Levent Ertaul, Manpreet Kaur, Venkata Arun Kumar R Gudise, Scrypt adalah hashing algorithm yang memanfaatkan fungsi derivasi kunci berdasarkan sandi [6]. Ini menghasilkan vektor besar string bit pseudorandom. Dibutuhkan banyak memori dan biaya CPU. Banyak nomor pseudorandom dihasilkan dalam seluruh proses yang disimpan dalam memori akses acak sehingga menempati ruang memori yang sangat besar. Ini dianggap sebagai algoritma yang mahal karena setiap elemen yang dihasilkan selama waktu hashing membutuhkan lebih banyak memori dan komputasi. Ini sangat aman karena sangat sulit bagi penyerang untuk memecahkan pesan berciri ini karena kurangnya sumber daya dan memori.

Metodologi Penelitian

Langkah pertama dalam melakukan penelitian ini adalah mencari jurnal-jurnal dan dokumentasi maupun publikasi elektronik yang berkaitan dengan topik SHA256, Scrypt dan juga Blockchain kemudian dari sumber-sumber referensi tersebut dilakukan penyusunan design program untuk membuktikan hipotesa yang dibuat yang nantinya dilakukan testing dan dilakukan Analisa terhadap data yang didapatkan.

Proyek ini akan membandingkan kemungkinan Algoritma SHA256 dan Scrypt Algoritma untuk menemukan blok baru. Program sumber yang digunakan untuk menjalankan tes adalah Bitcoin Core pada SHA256 dan Litecoin Core pada Scrypt. Untuk menyelesaikan tes, Penulis menggunakan 4 CPU yang berbeda untuk menjalankan tes.

Untuk membandingkan SHA256 dan Scrypt, Penulis akan mencoba untuk menguji dalam beberapa kondisi yang berbeda seperti:

1. Waktu Hashing: proses hashing akan dilakukan 3 kali untuk setiap algoritma dengan waktu hashing yang berbeda, yang pertama adalah 10 jam hashing, yang kedua adalah 3 jam hashing, dan yang terakhir adalah 1 jam hashing.
2. Test System: proses hashing akan dilakukan pada 4 CPU yang berbeda, yang pertama adalah Intel Pentium G4560, yang kedua adalah AMD Ryzen 3 1300X, yang ketiga adalah Intel Core i5-4200H, dan yang terakhir adalah AMD Ryzen 7 1800X.

Dari pengujian tersebut, Penulis akan mencatat berapa banyak percobaan pada setiap pengujian yang menggunakan pengukur hash dan berapa banyak blok baru yang ditemukan selama pengujian dengan menggunakan bahasa pemrograman awk dan akan digunakan untuk menghitung kemungkinan menemukan blok baru.

Hasil dan Pembahasan

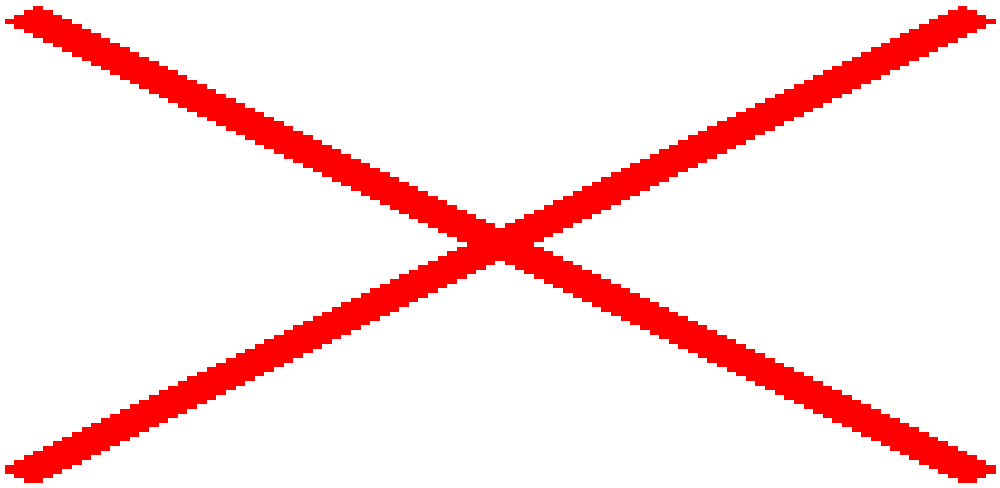
Penulis menjalankan pengujian pada 4 Sistem Tes yang berbeda untuk membandingkan kemungkinan menemukan blok baru pada setiap algoritma, semua tes direkam dengan simplescreenrecorder pada Linux Ubuntu 16.04 dan masuk ke file `debug.log`. Dari setiap file log, Penulis menggunakan awk untuk mengumpulkan informasi dari semua file log.

Dari file debug penulis tahu bahwa penemuan blok baru ditandai dengan 'proof-of-work found', dan upaya dapat dihitung dengan pengukur hash dalam kh / s. Dan kemudian Pengarang mengumpulkan data ini:

Tabel blok baru yang ditemukan selama pengujian, direkam menggunakan debug.log dan diproses menggunakan utilitas awk pada sistem Linux:

Tabel 1: Tabel blok baru

CPU	New Block on SHA256 in 10 hours	New Block on SHA256 in 3 hours	New Block on SHA256 in 1 hours	New Block on Scrypt in 10 hours	New Block on Scrypt in 3 hours	New Block on Scrypt in 1 hours
G4560	31	13	4	265	92	34
i5 4200H	35	10	4	203	68	26
R3 1300X	190	58	17	375	107	23
R7 1800X	864	227	76	1414	382	136



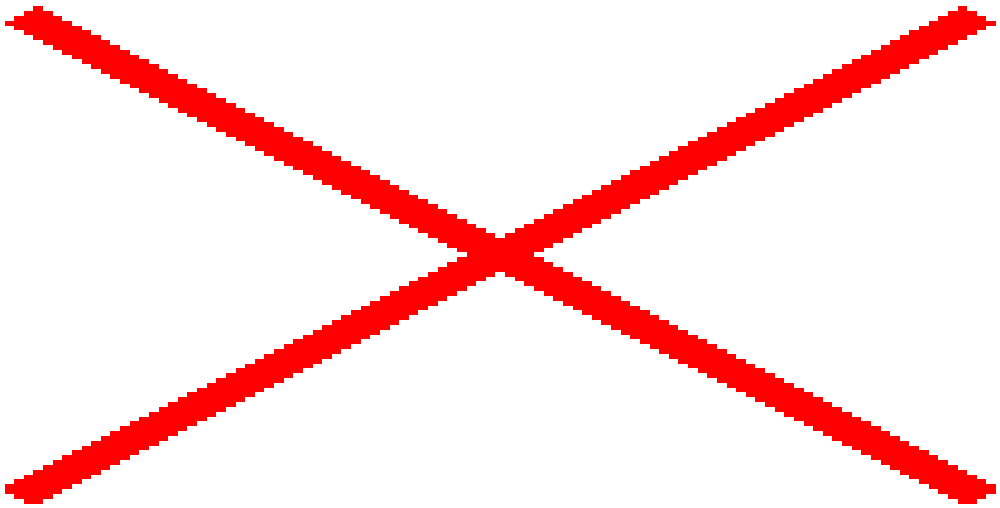
Gambar 1: Grafik blok baru

Dari data di atas meja Anda dapat melihat perbedaan besar antara dua algoritma dalam 24 skenario pengujian yang berbeda, algoritma Scrypt dapat menemukan lebih banyak blok baru daripada algoritma SHA256 di setiap tes. Dari analisis saya itu terjadi karena algoritma Scrypt memiliki banyak parameter biaya yang telah ditetapkan dan itu membuat algoritma scrypt memiliki target hashing lebih spesifik daripada SHA256.

Tabel hash meter yang direkam selama tes, dikumpulkan dari debug.log dan diproses menggunakan utilitas awk pada sistem Linux:

Tabel 2: Table hashmeter

CPU	Hashmeter on SHA256 in kh/s	Hashmeter on SHA256 in kh/s	Hashmeter on SHA256 in kh/s	Hashmeter on Scrypt in kh/s	Hashmeter on Scrypt in kh/s	Hashmeter on Scrypt in kh/s
G4560	4619	4614	4619	9	9	9
i5 4200H	3516	3910	3905	6.55	6.5	6.7
R3 1300X	21305	20809	20753	10	10	10
R7 1800X	101111	88624	88271	41	36	36



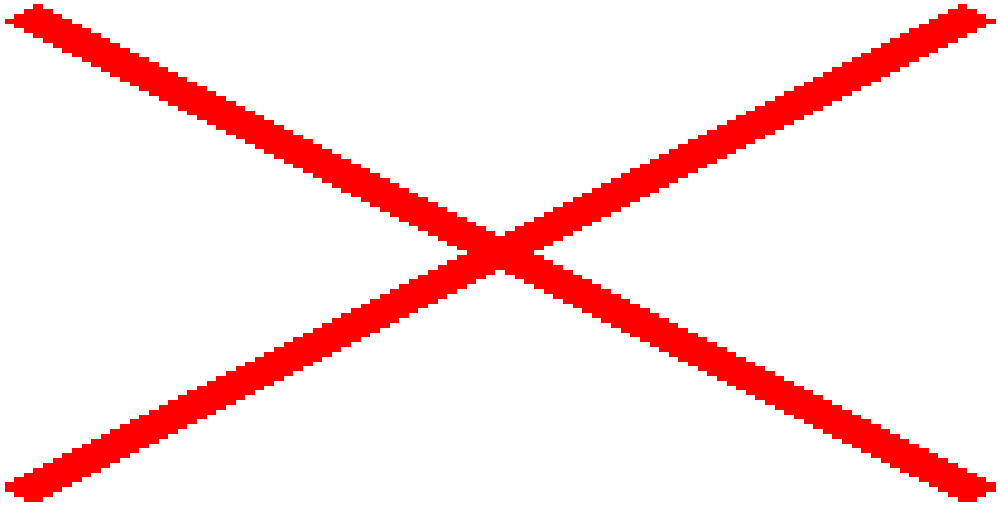
Gambar 2: Grafik hashmeter

Dari data Anda hampir tidak dapat melihat kesenjangan besar antara dua algoritma, sistem pengujian dapat memiliki lebih banyak blok pada SHA256 daripada algoritma Scrypt di setiap sistem uji. Dari analisis saya itu terjadi karena scrypt memiliki proses yang lebih rumit daripada SHA256, bahkan algoritma Scrypt dapat berisi algoritma hashing lain dalam prosesnya juga. Itu sebabnya itu lebih lambat daripada algoritma Scrypt.

Tabel percobaan yang dihitung dengan mengalikan hashmeter dan jumlah detik dalam setiap skenario pengujian yang berjalan selama pengujian, diproses menggunakan utilitas awk pada sistem Linux:

Tabel 3: Tabel jumlah percobaan

CPU	Attempt on SHA256 in 10 hours	Attempt on SHA256 in 3 hours	Attempt on SHA256 in 1 hours	Attempt on Scrypt in 10 hours	Attempt on Scrypt in 3 hours	Attempt on Scrypt in 1 hours
G4560	1.66E+11	49831200000	16628400000	324000000	97200000	32400000
i5 4200H	126576000000	42228000000	14058000000	235800000	70200000	24120000
R3 1300X	766980000000	224737000000	74710800000	360000000	108000000	36000000
R7 1800X	3640000000000	957139000000	317776000000	1476000000	388800000	129600000



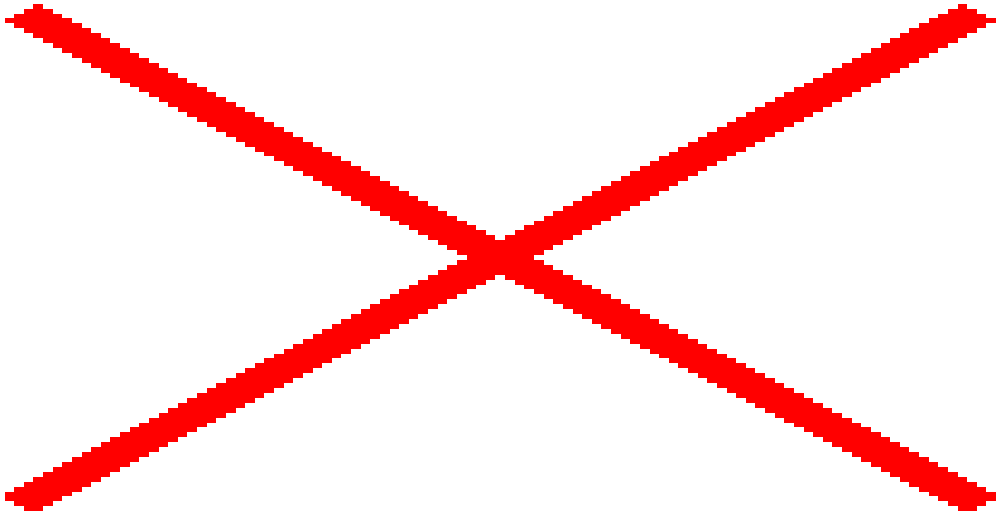
Gambar 3: Grafik jumlah percobaan

Dari data yang Anda ketahui bahwa upaya pada SHA256 jauh lebih banyak daripada pada algoritme Scrypt, itu terjadi karena pengukur hash pada SHA256 jauh lebih cepat daripada algoritma Scrypt dan itu berarti SHA256 dapat melakukan lebih banyak upaya daripada algoritma Scrypt. Bentuk analisis saya, hanya dengan melihat hash meter Anda tahu bahwa itu terjadi karena SHA256 memiliki proses hashing lebih cepat daripada algoritma Scrypt.

Tabel probabilitas yang dihitung dari semua data yang dikumpulkan dari tes dan diproses menggunakan LibreOffice Calc pada sistem Linux:

Tabel 4: Tabel probabilitas ditemukannya blok baru

CPU	SHA256 Probability in 10 hours	SHA256 Probability in 3 hours	SHA256 Probability in 1 hours	Scrypt Probability in 10 hours	Scrypt Probability in 3 hours	Scrypt Probability in 1 hours
G4560	0	0	0	8.17901E-05	0	0
i5 4200H	0	0	0	0	0	0
R3 1300X	0	0	0	0	0	0
R7 1800X	0	0	0	0	0	0



Gambar 4: Grafik probabilitas ditemukannya blok baru

Dari data pada tabel ini Anda tahu bahwa algoritma Scrypt memiliki probabilitas yang sedikit lebih tinggi daripada algoritma SHA256 di setiap skenario pengujian. Dari analisis saya itu semua terjadi karena algoritma Scrypt memiliki parameter yang lebih standar seperti parameter biaya CPU, parameter biaya memori, parameter Paralelisasi, panjang Output dll yang membuatnya memiliki proses hashing lebih spesifik dan mencegah algoritma membuang sumber dayanya.

Kesimpulan

Dari semua tes yang dilakukan, dapat disimpulkan bahwa algoritma Scrypt dapat menemukan lebih banyak blok baru daripada algoritma SHA256 meskipun SHA256 dapat melakukan lebih banyak upaya di detik tetapi jumlah blok baru yang ditemukan kurang dari algoritma Scrypt. Oleh karena itu, penulis menyimpulkan bahwa Scrypt memiliki probabilitas yang lebih tinggi dalam menemukan blok baru daripada algoritma SHA256 pada teknologi Blockchain.

Bagaimanapun, penulis menyarankan untuk penelitian berikutnya untuk melakukan penelitian dalam hal keamanan pada algoritma yang terkandung dalam teknologi blockchain. Jadi, dapat dipastikan bahwa algoritma tidak hanya memiliki probabilitas

tinggi untuk menemukan blok baru, tetapi juga menyediakan keamanan data dalam teknologi Blockchain.

Daftar Pustaka

[1] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A Scalable Blockchain Protocol," *CoRR*, vol. abs/1510.02037, 2015, [Online]. Available: http://arxiv.org/abs/1510.02037.

[2] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "BLOCKCHAIN – THE GATEWAY TO TRUST-FREE CRYPTOGRAPHIC TRANSACTIONS," *Research Papers*, May 2016, [Online]. Available: https://aisel.aisnet.org/ecis2016_rp/153.

[3] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, "On the Privacy Provisions of Bloom Filters in Lightweight Bitcoin Clients," in *Proceedings of the 30th Annual Computer Security Applications Conference*, New York, NY, USA, 2014, pp. 326–335, doi: 10.1145/2664243.2664267.

[4] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2016, pp. 3–16, doi: 10.1145/2976749.2978341.

[5] J. Alwen, B. Chen, K. Pietrzak, L. Reyzin, and S. Tessaro, "Scrypt Is Maximally Memory-Hard," in *Advances in Cryptology – EUROCRYPT 2017*, Cham, 2017, pp. 33–62.

[6] L. Ertaul and M. Kaur, "Implementation and Performance Analysis of PBKDF 2, Bcrypt, Scrypt Algorithms varunkrg."

PENCARIAN POHON PERENTANG MINIMUM PADA JARINGAN LISTRIK BANGUNAN DENGAN MENGGUNAKAN ALGORITMA KRUSKAL

Yuniarto Bagas Wibisono[1], YB. Dwi Setianto[2]

[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Katholik Soegijapranata

[1]14k10048@student.unika.ac.id, [2]setianto@unika.ac.id

Abstract

The minimum spanning tree can be used to solve the problems in the graph. An example of a graph problem can be found in a building's electrical network.This project discusses the analysis of Kruskal algorithm on the building plan. The building plan that has been obtained is then done by mapping with switches and socket as nodes and cables as edges. After testing the obtained data, it can be concluded that the Kruskal algorithm can be used to determine the minimum spanning tree.

Keywords: *minimum spanning tree, kruskal, electrical*

Pendahuluan

Dalam membangun sebuah rumah ada banyak faktor yang harus diperhatikan, salah satunya adalah ketersediaan jaringan listrik. Pemasangan jaringan listrik yang tidak efisien tentu akan menimbulkan pembengkakan biaya. Dengan berkembangnya teknologi, pemasangan jaringan listrik harus dilakukan secara efisien dan optimal sehingga dapat menekan anggaran instalasi supaya dapat di alokasikan ke bidang yang lain.

Teori graf dapat digunakan untuk menyelesaikan masalah ini. Jaringan listrik yang sudah terpasang dapat direpresentasikan ke dalam bentuk graf berbobot, terhubung, dan tak berarah (*weighted, connected, and undirect graph*). Panjang kabel listrik yang terpasang dapat diminimalkan dengan menggunakan algoritma Kruskal. Setiap langkah pada algoritma ini akan membentuk tree, yang pada akhirnya tujuan dari algoritma ini adalah untuk menentukan *cost* terkecil dari pohon rentang minimum (*Minimum Spanning Tree*).

Dengan adanya penjelasan kasus diatas, maka project ini menghasilkan sebuah aplikasi yang mampu menentukan keoptimalan panjang kabel pada desain jaringan listrik di sebuah bangunan.

Landasan Teori

Pada penelitian tentang yang ditulis oleh Wattimena dan Lawalata, menjelaskan tentang bagaimana mengatasi permasalahan jaringan air di pemasangan pipa untuk dialirkan ke rumah-rumah [1]. Dengan menggunakan algoritma Kruskal, maka permasalahan yang terjadi dapat di selesaikan secara optimal. Panjang pipa yang

digunakan dalam data sampel penelitian ini yaitu 1448 meter, sedangkan panjang pipa yang dihasilkan oleh sistem adalah 1026 meter.

Wamiliana, Kurniawan, dan Shavitri menjelaskan tentang pengembangan algoritma untuk optimalisasi sumber daya [2]. Dalam penelitian ini, mereka membandingkan kinerja antara algoritma Prim, algoritma Kruskal dan algoritma Sollin. Meskipun langkah-langkah yang dilakukan setiap algoritma berbeda, penelitian ini menghasilkan kesimpulan bahwa ketiga algoritma ini menghasilkan *output* yang sama.

Penelitian yang dilakukan oleh Anggraeni menjelaskan bahwa untuk menekan anggaran perbaikan jalan raya, pemerintah harus membuat kebijakan dengan memilih ruas jalan mana yang akan diperbaiki [3]. Setidaknya harus ada ruas jalan yang menghubungkan antar satu kabupaten dengan yang lainnya supaya akses antar kabupaten tidak terputus. Penelitian ini menggunakan algoritma Sollin dengan menghasilkan ruas jalan yang optimal untuk diperbaiki sepanjang 1251 kilometer.

Algoritma Prim dan Kruskal mampu digunakan untuk mencari pohon rentang minimum untuk graf berbobot. Pernyataan ini dibuktikan oleh Latifah dan Sugiharti dengan penelitian pada jaringan distribusi air [4]. Pada penelitian ini, pencarian pohon rentang minimum menggunakan algoritma Prim dan Kruskal dengan bantuan program Matlab.

Dalam project ini menerapkan algoritma Kruskal, seperti yang telah diterapkan oleh Wattimena, Lawalata, dan Umi. Algoritma ini digunakan untuk memeriksa jalur listrik utama yang tersedia pada sebuah bangunan sehingga dapat menghasilkan jalur optimal dengan syarat semua titik terhubung (*minimum spanning tree*). Data jalur listrik yang tersedia disimpan dalam program. Project ini menggunakan bahasa pemrograman Python untuk mengimplementasikan algoritma Kruskal.

Metodologi Penelitian

Penelitian ini menggunakan data berupa denah bangunan. Data yang digunakan adalah denah CV Inti Jaya Printing yang didapat dari PT Bina Putera Perkasa Semarang.

Beberapa langkah yang dilakukan dalam penelitian ini antara lain adalah:

1. Pengambilan data berupa denah sebuah bangunan.
2. Penerapan algoritma Kruskal dalam pemodelan jaringan listrik yaitu sebagai berikut:
    a) Menentukan saklar dan stopkontak sebagai titik (node) dan kabel sebagai sisinya (edge)
    b) Penentuan minimum spanning tree dilakukan dengan cara memberikan asumsi kemungkinan jalur yang akan diperiksa oleh Kruskal.
    c) Menginputkan data-data panjang sisi kedalam program.
    d) Mengurutkan sisi-sisi pada graf G mulai dari sisi tependek hingga sisi yang paling panjang.

e) Menggagalkan sisi yang membentuk sirkuit hingga tersisa sisi dengan jumlah (*n*-1) dimana *n* merupakan jumlah titik.

f) Mendapatkan hasil minimum spanning tree dari jaringan listrik yang telah dibuat dengan menggunakan algoritma Kruskal.

3. Desain diagram alir program

Peran diagram alir (*flowchart*) sangatlah penting dalam membuat sebuah program. *Flowchart* digunakan untuk menggambarkan, menyederhanakan, rangkaian proses atau prosedur sehingga mudah dipahami.

4. Implementasi

Diagram alir yang telah di desain kemudian di implementasikan ke dalam program dengan menggunakan Python.

5. Pengujian program

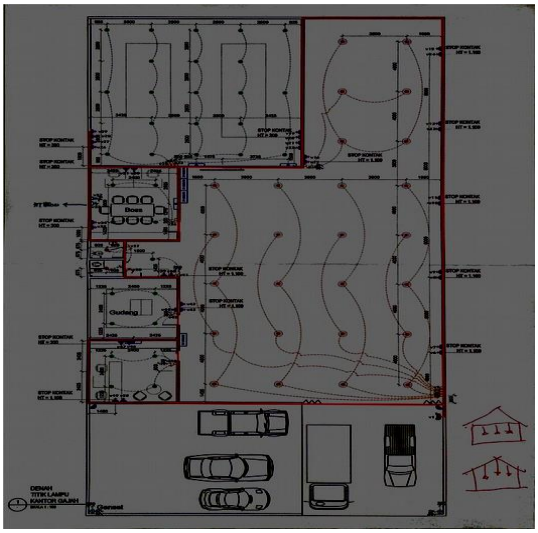
Setelah melalui tahap implementasi, program yang telah dibuat harus melalui tahap *testing*. Fungsi dari tahap ini adalah untuk memastikan apakah masih ada error dalam program dan memastikan bahwa algoritma dapat berjalan dengan benar.

6. Penarikan kesimpulan

Penarikan kesimpulan dilakukan pada akhir penelitian dan berdasarkan hasil dari tahap pengujian program.

Hasil dan Pembahasan

Pengujian pada project ini dilakukan pada 2 buah denah yang telah didapat. Denah tersebut akan diproses sehingga menghasilkan graf yang akan diperiksa oleh algoritma Kruskal. Pengujian diawali dengan pemberian asumsi jalur yang akan diperiksa oleh Kruskal.



Gambar 1: Sampel Denah Bangunan

Gambar diatas adalah denah bangunan yang sudah diberi jalur. Selanjutnya data nodes dan edges dari denah tersebut disimpan ke dalam program. Data nodes dan edges dapat dilihat pada tabel 1.

Tabel 1: Data Nodes dan Edges

Nodes & Edges	Bobot	Nodes & Edges	Bobot
v1 v2	150	v26 v27	40
v1 v50	1870	v26 v32	410
v2 v3	20	v27 v28	40
v3 v4	20	v28 v29	40
v4 v5	20	0	480
v5 v6	280	0	40
v6 v7	40	v31 v32	450
v7 v8	550	v31 v26	350
v8 v9	40	v32 v33	40
v9 v10	550	v33 v34	150
v10 v11	40	v34 v35	40
v11 v12	550	v35 v37	340
v12 v13	40	v36 v37	520
v13 v14	550	v37 v38	180
v14 v15	40	v38 v39	370
v15 v19	1860	0	40
v16 v17	40	0	30
v17 v18	40	v41 v42	250
v18 v23	720	v42 v43	40
v19 v20	40	v43 v44	40
v20 v21	40	v44 v45	30
v22 v16	70	v45 v46	480
v22 v23	850	v45 v48	400
v23 v24	20	v46 v47	40
v24 v25	20	v47 v49	790
v25 v30	200	0	40
v25 v26	350	0	600

Data diatas kemudian diproses oleh program. Program akan membaca data yang tersimpan untuk menghasilkan graf. Berdasarkan graf yang telah dibentuk, program mendapatkan hasil panjang kabel 10.470 cm. Hasil yang sama juga didapatkan ketika penghitungan Kruskal dilakukan secara manual tanpa bantuan program, yaitu sepanjang 10.470 cm. Rincian jalur yang dipilih oleh program dapat dilihat pada Gambar 2.

```
================================================
Total panjang kabel = 10470 centimeter /
Dengan rincian jalur sebagai berikut :
[('v2', 'v3', 20), ('v3', 'v4', 20), ('v4', 'v5', 20), ('v23', 'v24', 20), ('v24', 'v25', 20), ('v
43', 'v44', 20), ('v40', 'v41', 30), ('v44', 'v45', 30), ('v6', 'v7', 40), ('v8', 'v9', 40), ('v10
', 'v11', 40), ('v12', 'v13', 40), ('v14', 'v15', 40), ('v16', 'v17', 40), ('v17', 'v18', 40), ('v
19', 'v20', 40), ('v20', 'v21', 40), ('v21', 'v22', 40), ('v26', 'v27', 40), ('v27', 'v28', 40), (
'v28', 'v29', 40), ('v30', 'v31', 40), ('v32', 'v33', 40), ('v34', 'v35', 40), ('v39', 'v40', 40),
 ('v42', 'v43', 40), ('v46', 'v47', 40), ('v49', 'v50', 40), ('v22', 'v16', 70), ('v1', 'v2', 150)
, ('v33', 'v34', 150), ('v37', 'v38', 180), ('v25', 'v30', 200), ('v41', 'v42', 250), ('v5', 'v6',
 280), ('v35', 'v37', 340), ('v25', 'v26', 350), ('v38', 'v39', 370), ('v45', 'v48', 400), ('v26',
 'v32', 410), ('v30', 'v36', 480), ('v45', 'v46', 480), ('v7', 'v8', 550), ('v9', 'v10', 550), ('v
11', 'v12', 550), ('v13', 'v14', 550), ('v50', 'v48', 600), ('v18', 'v23', 720), ('v15', 'v19', 18
60)]
```

Gambar 2: Hasil Algoritma Kruskal

Berdasarkan hasil jalur yang telah didapatkan, maka visualisasi jalur dapat dilakukan. Visualisasi pada project ini masih dilakukan secara manual. Hasil jalur yang telah di visualisasi ke dalam denah sampel bangunan dapat dilihat pada Gambar 3.

Gambar 3:

Kesimpulan

Berdasarkan hasil tes yang telah dilakukan, algrotima Kruskal dapat digunakan untuk menemukan minimum spanning tree pada jaringan listrik utama dari sebuah bangunan. Hasil perhitungan yang dihasilkan oleh program sama dengan hasil yang didapat ketika melakukan perhitungan manual tanpa bantuan program.

Daftar Pustaka

[1] A. Z. Wattimena and S. Lawalatta, “APLIKASI ALGORITMA KRUSKAL DALAM PENGOTIMALAN PANJANG PIPA,” *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 7, no. 2, Art. no. 2, 2013.

[2] D. Kurniawan, W. Wamiliana, and C. S. N. Fauzi, “perbandingan kompleksitas algoritma prim, algoritma kruskal, dan algoritma sollin untuk menyelesaikan masalah minimum spanning tree,” *Jurnal Komputasi*, vol. 2, no. 1, Art. no. 1, Sep. 2016, doi: 10.23960/komputasi.v2i1.1005.

[3] W. Anggraeni, “APLIKASI ALGORITMA SOLLIN DALAM PENCARIAN POHON PERENTANG MINIMUM PROVINSI JAWA TENGAH,” *Faktor Exacta*, vol. 8, no. 4, Art. no. 4, Dec. 2015, doi: 10.30998/faktorexacta.v8i4.508.

[4] U. Latifah and E. Sugiharti, “PENERAPAN ALGORITMA PRIM DAN KRUSKAL PADA JARINGAN DISTRIBUSI AIR PDAM TIRTA MOEDAL CABANG SEMARANG UTARA,” *Unnes Journal of Mathematics*, vol. 4, no. 1, Art. no. 1, May 2015, doi: 10.15294/ujm.v4i1.7418.

COMPARISON OF STOCK PRICE PREDICTION ACCURACY WITH VARIATION NUMBER OF HIDDEN LAYER CELL IN BACKPROPAGATION ALGORITHM

Doohan Kristiawan[1], Rosita Herawati[2]

[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata

[1]doohanhahaha@gmail.com, [2]rosita@unika.ac.id

Abstract

Nowadays there are various ways to invest money. The capital market becomes a tantalizing choice for potential investors. The profit from this investment is obtained from the difference between the selling and buying price of the shares. Created program is expected to provide an overview comparison of prediction accuracy from different hidden layer cells amount. Using historical data from yahoo finance to get stock prices in last 12 months. The data is processed using backpropagation algorithm and the process is done by the number of different hidden layer. Eventually the program will give the stock price predictions from four different number of hidden layer. Of all the proceeds will be matched to determine which one has the most precise accuracy.

Keywords: *backpropagation, neural network, artificial intelligence*

Pendahuluan

Banyak cara yang bisa dilakukan untuk berinvestasi. Kemajuan teknologi membuatnya menjadi lebih mudah. Salah satu investasi yang paling menarik adalah saham. Investasi saham tergantung pada pergerakan harga. Investor akan membeli saham tersebut saat harga rendah dan menjualnya ketika harganya tinggi. Namun ada banyak faktor yang mempengaruhi pergerakan harga saham seperti rumor dan sentimen pasar, kebijakan pemerintah, fluktuasi nilai tukar mata uang asing, dll. Pergerakan harga saham yang cenderung tidak stabil membuat banyak orang mencari solusi untuk menganalisa pergerakan. Padahal data harga saham yang bisa dianalisis itu tersedia di finance.yahoo.com. Dengan mengolah data di sana, siapapun bisa melakukan analisa.

Ada dua metode untuk menganalisa prediksi saham yaitu fundamental dan teknis. Analisis fundamental meliputi kinerja perusahaan, analisis industri, analisis persaingan usaha, analisis ekonomi dan makro mikro. Analisis fundamental adalah cara yang sulit karena butuh banyak informasi yang tidak bisa didapatkan oleh masyarakat. Karena analisa fundamental memang sulit, maka analisa teknikal menjadi solusi terbaik bagi investor. Analisis teknis meliputi moving average convergence divergence, indeks kekuatan relatif, stochastic oscillator dan lain-lain. Karena banyaknya data dan parameter yang akan diuji, analisa teknikal menjadi sangat sulit dilakukan secara manual. Salah satu cara untuk mengatasi masalah ini adalah dengan menggunakan jaringan syaraf tiruan.

Jaringan syaraf tiruan memiliki algoritma yang disebut backpropagation. Algoritma ini bisa memberikan prediksi berdasarkan data di masa lalu. Sangat cocok untuk prediksi

saham karena ada data historis yang bisa diolah. Algoritma backpropagation dapat belajar dari kesalahan dan memperbaiki nilai untuk memberikan hasil yang paling optimal.

Algoritma backpropagation membutuhkan data saham pada masa lampau untuk diproses terlebih dahulu. Data masa lampau tersebut tersedia di finance.yahoo.com bisa diambil dengan teknik webscraping. Webscraping adalah cara untuk mengambil data yang terdapat di situs web. Data yang diambil adalah harga saham selama 6 bulan terakhir. Kemudian akan dianalisis dengan 4 cara yang berbeda, dengan jumlah sel lapisan tersembunyi yang berbeda mulai dari 2, 4, 6 dan 8 sel. Lapisan sel tersembunyi adalah suatu tempat yang memproses bobot dari input sehingga menghasilkan suatu prediksi. Setiap susunan lapisan tersembunyi akan memberikan hasil prediksi dan hasilnya akan dibandingkan dengan harga realita. Hasil analisis ini dapat digunakan untuk memprediksi harga saham.

Landasan Teori

Investor saham memiliki masalah dalam menganalisa pergerakan harga saham dan memprediksi harga. Ada dua cara untuk menganalisa pergerakan harga saham dengan analisa fundamental dan analisa teknikal. Analisis fundamental membutuhkan banyak data yang tidak bisa diakses oleh masyarakat, sedangkan analisa teknikal sangat sulit untuk dihitung secara manual. Oleh karena itu, penulis akan membuat sebuah program yang bisa memprediksi harga saham pada masa lampau. Program ini akan mudah digunakan dan jauh lebih cepat daripada harus menghitung secara manual.

Program ini akan dibuat dengan menggunakan algoritma backpropagation yang merupakan sub bab dari jaringan syaraf tiruan. Algoritma backpropagation dapat memberikan prediksi berdasarkan data history. Data historis tersedia di finance.yahoo.com dan dapat diambil dengan menggunakan webscrapping.

Langkah – langkah yang akan dilakukan:

1. Mengambil data historis dari finance.yahoo.com

   Pada jurnal berjudul “Penerapan Metode Jaringan Syaraf Tiruan Backpropagation Untuk Meramalkan Harga Saham (IHSG)” karangan Andri Triyono dkk, menjelaskan bahwa data saham dapat diambil dari finance.yahoo.com [1]. Data saham yang akan diambil adalah TLKM, HMSP, BBCA, BBNI dan ASII. Data akan diambil mulai Januari 2016 sampai November 2017. Data dari Januari sampai Desember 2016 akan digunakan sebagai data pelatihan dan Januari - November 2017 akan digunakan sebagai data testing. Data akan disimpan dalam file teks.

2. Normalisasi data

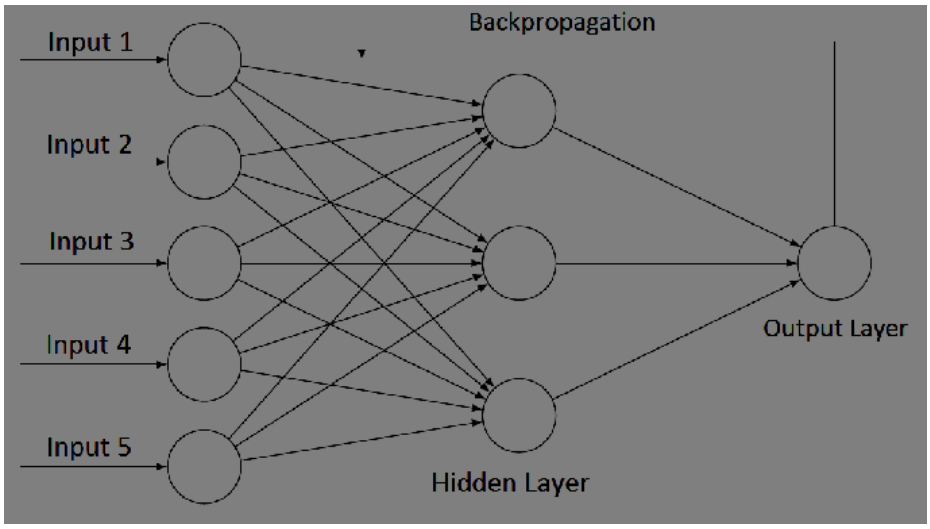
   Data yang sudah didapat akan dinormalisasi. Pada jurnal berjudul “Prediksi Harga Saham Menggunakan Jaringan Syaraf Tiruan Backpropagation” karangan Siti Amiroch, dijelaskan tujuan normalisasi adalah menyamarakan data yang berbeda besarannya agar menjadi setara dengan cara mengubah semua bilangan menjadi kisaran 0.1 sampai 0.8 [2].

x : data asli

MinValue : nilai terendah dari kumpulan data.

MaxValue : nilai tertinggi dari kumpulan data.

3. Proses Backpropagation



Gambar 1 : Proses Backpropagation

Pada buku yang berjudul “Kecerdasan Buatan” karangan T Sutojo dkk, dijabarkan langkah – langkah algoritma backpropagation untuk memperoleh suatu hasil prediksi [3].

Langkah – langkah tersebut yaitu :

Langkah 0: Inisialisasi semua bobot dengan bilangan acak kecil.

Bobot bias masukan (v0j) = bilangan acak dari -ß dan ß.

Masukan berat (Vij) = bilangan acak -0,5 dan 0,5.

Bobot bias hidden layer (W0k) = bilangan acak dari -1 dan 1

Berat hidden layer (Wjk) = bilangan acak dari -1 dan 1.

Langkah 2: Untuk setiap data training lakukan langkah 3-8.

Langkah 3 : Hitung semua outputs di hidden unit (Zj, j = 1, .., p)

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$$
$$z_j = f(z\_in_j)$$

Langkah 4 : Hitung fungsi aktivasi pada hidden layer

Langkah 5 : Operasi pada output layer.

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$$
$$y_k = f(y\_in_k)$$

Langkah 6 : Hitung fungsi aktivasi pada output layer

Langkah 7 : Hitung δ pada output unit berdasarkan error pada tiap unit of output

$$\delta_k = (t_k - y_k) f'(y\_in_k)$$

Hitung selisih bobot hidden and bobot hidden menggunakan learning rate α

$$\Delta w_{jk} = \alpha \delta_k z_j$$
$$\Delta w_{0k} = \alpha \delta_k$$

Langkah 8 : Hitung δ pada hidden units berdasarkan error pada tiap hidden unit Zj

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$$
$$\delta_j = \delta\_in_j f'(z\_in_j)$$

Hitung berbedaan bobot nilai input dan bobot input bias dengan learning rate α

$$\Delta v_{ij} = \alpha \delta_j x_i$$
$$\Delta v_{0j} = \alpha \delta_j$$

Langkah 9 : Hitung semua perubahan bobot

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk}$$
$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij}$$

Langkah 10 : Tes kondisi perhentian berdasarkan square error

Langkah 11 : Ulangi langkah 3-10 sampai squared error < 0.01

Langkah 12 : Hitung dengan jumlah cell lain

4. Proses Pengujian

Hasil backpropagation akan dibandingkan dengan data historycal pada bulan Januari - November 2017. Pada jurnal berjudul "Prediksi Pergerakan Harga Saham Menggunakan Metode Backpropagation Neural Network" karangan Rio Bayu Afrianto, dijelaskan bahwa untuk mengecek keakuratan setiap prediksi maka hasil akan dibandingkan dengan hasil yang lain untuk menentukan jumlah sel lapisan tersembunyi mana yang bisa memberikan akurasi paling akurat [4].

Metodologi Penelitian

Langkah-langkahnya dijalankan satu per satu dan pada akhir penelitian akan didapatkan data perbandingan. Dari data perbandingan dapat disimpulkan dimana lapisan tersembunyi paling akurat untuk prediksi saham.

Langkah yang dilakukan:

1. Mempelajari jurnal dan buku

Jurnal dan buku dipelajari terlebih dahulu untuk mendukung penelitian. Ada 3 jurnal dan 1 buku yang diteliti. Dari berbagai jurnalis peneliti akan belajar dan mencoba menerapkannya pada penelitian ini sambil memberikan sudut pandang yang berbeda sehingga bisa memberikan gambaran yang berbeda mengenai proses pada jaringan syaraf tiruan.

2. Menganalisa masalah

Analisis semua masalah yang akan dipecahkan oleh program. Hasil analisis akan digunakan untuk membuat desain proyek, sehingga implementasinya akan sesuai dengan tujuan utama dan tetap efisien untuk memenuhi tenggat waktu yang disepakati.

3. Pengambilan data dan proses coding

Dalam penelitian ini harga saham yang akan digunakan adalah PT Telekomunikasi Indonesia Tbk (TLKM), PT HM Sampoerna Tbk (HMSP), PT Bank Central Asia Tbk (BBCA), PT Bank Negara Indonesia Tbk (BBNI) dan PT Astra International Tbk (BBNI) ASII). Saham tersebut dipilih untuk masuk dalam kategori lima saham utama di Indonesia. Pasar saham buka mulai Senin sampai Jumat sehingga dalam 12 bulan mulai Januari - Desember 2016 akan diperoleh sebanyak 151 data dan total 5 perusahaan adalah 755 data. Dari 12 bulan data yang diperoleh akan digunakan untuk data training dan testing. Proses pengambilan data akan menggunakan metode webscraping menggunakan Java HTML parser (JSOUP). Setelah diperoleh data akan disimpan dalam file teks. Proses pengkodean akan menggunakan bahasa pemrograman java.

4. Laporan penelitian

Langkah ini melaporkan proses implementasi yang dapat memberikan prediksi harga saham. Prediksi harga akan dikumpulkan dan dibandingkan. Hal ini juga dilaporkan bahwa jumlah sel lapisan tersembunyi dapat memberikan akurasi yang paling akurat.

Hasil dan Pembahasan

Aplikasi ini dibuat dengan menggunakan bahasa pemprograman Java. Linked list sebagai struktur data digunakan untuk menyimpan data saham. Linked list dipilih karena stok data selalu tumbuh. Algoritma backpropagation digunakan oleh variasi jumlah sel lapisan tersembunyi yang berbeda. Variasi jumlah sel lapisan tersembunyi adalah 2, 4, 6 dan 8. Semua perbedaan lapisan tersembunyi akan menghasilkan prediksi harga yang berbeda. Beberapa percobaan akan dilakukan dengan menggunakan aplikasi ini. Hasil percobaan ini akan memberikan prediksi yang paling optimal dan akurat.

Aplikasi ini akan memproses data history stock yang diambil dari finance.yahoo.com. Data stok yang akan diambil adalah TLKM, HMSP, BBCA, BBNI dan ASII. Data akan diambil mulai Januari 2016 sampai November 2017. Data dari Januari sampai Desember 2016 akan digunakan sebagai data pelatihan dan Januari - November 2017 akan digunakan sebagai data testing. Data stok diambil dan disimpan ke dalam file teks.

Data yang telah disimpan akan dinormalisasi sebelum digunakan dalam proses perhitungan. Data harus dinormalisasi karena data volume puluhan juta namun harga terbuka, tinggi, rendah dan menyesuaikan data hanya dalam ribuan. Jadi normalisasi digunakan untuk menyamakan nilai data yang berbeda. Setelah data dinormalisasi maka tahap "feed forward" akan terus berlanjut. Tahap umpan maju akan menghasilkan prediksi sementara. Prediksi sementara dibandingkan dengan harga asli dan selisihnya akan dihitung. Perbedaan tersebut digunakan untuk proses backpropagation sehingga bobotnya akan disesuaikan. Kemudian dengan bobot yang disesuaikan, proses diulang sampai mencapai target zaman.

Mengambil Data Saham

Tahap pertama dari aplikasi ini adalah mengambil data saham dari finance.yahoo.com. Teknik pengambilan data dari internet disebut webscraping. Dalam bahasa pemrograman java ada sebuah perpustakaan bernama JSOUP yang bisa digunakan untuk webscraping. Sebelum mengambil data, aplikasi akan memeriksa data saham yang telah tersimpan dalam file teks dan melihat data terakhir yang dimilikinya. Kemudian lakukan webscrap untuk mengambil data baru yang belum tersimpan dalam file teks. Data yang lebih baru ditambahkan ke dalam file teks.

Normalisasi

Data dalam linked list akan masuk ke dalam proses normalisasi data. Normalisasi digunakan agar data yang berbeda menjadi lebih menyamakan kedudukan dan memiliki rentang 0.1 – 0.8. Normalisasi membutuhkan nilai minimum dan maksimum suatu data

yang harus dicari sebelum masuk dalam proses formula. Hasil dari proses ini adalah akan disimpan ke dalam linked list.

Feed Forward

Daftar linked yang dinormalisasi masuk ke dalam proses umpan maju. Pada awalnya ada beberapa bobot yang harus diinisiasi seperti masukan bobot ke lapisan tersembunyi, bobot bias ke lapisan tersembunyi, bobot lapisan tersembunyi ke lapisan keluaran dan bobot bias ke lapisan keluaran. Data berat secara acak diisi dengan kisaran -0,5 sampai 0,5. Bobot ini hanya sebagai angka awal saja, saat memasuki tahap Backpropagation maka bobotnya akan berubah sesuai perhitungan.

Tahap feed forward melakukan perhitungan operasi pada lapisan tersembunyi, fungsi aktivasi pada lapisan tersembunyi, operasi pada lapisan output dan fungsi aktivasi lapisan keluaran. Hasil prediksi akan muncul pada fungsi aktivasi lapisan output.

Proses pada tahap Feed Forward ini adalah menghitung input dengan bobot awal untuk menghasilkan prediksi harga sementara. Disebut prediksi sementara karena pada tahap selanjutnya prediksi ini akan diperbaiki lagi dengan hasil prediksi yang lebih baik. Kelima input berupa harga terbuka, tinggi, rendah, harga penyesuaian dan volume akan menghasilkan output yang harganya mendekati.

Menghitung Persentase Error

Setelah hasil prediksi muncul, hitunglah error rate dengan menggunakan rumus RMSE. Dari hasil RMSE yang bisa diketahui apakah datanya optimal atau tidak. Rumus untuk menghitung RMSE:

$$\mathrm{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$$

Ypred : Harga prediksi

Yref : Harga asli

N : Jumlah data

Setiap kali proses jaringan syaraf berjalan maka hasil yang diprediksi akan bertambah akurat antara harga asli dan kuadrat. Hasilnya akan terus ditambahkan sampai semua proses jaringan syaraf selesai dan hasilnya dibagi dengan jumlah data yang digunakan.

Perhitungan rumus RMSE adalah untuk mengetahui keakuratan prediksi. Semakin kecil nilai RMSE hasil prediksi yang lebih akurat dengan harga sebenarnya. Nilai RMSE dianggap kecil jika nilainya di bawah 0,01.

Proses Backpropagation

Tahap Backpropagation adalah memperbaiki nilai bobot agar lebih optimal dengan jaringan. Harga prediksi sementara dari tahap Feed Forward dibandingkan dengan harga

semula. Perbedaan harga untuk mengubah semua nilai bobot. Nilai bobot akan diperbarui sehingga bobot menjadi lebih akurat. Proses menyesuaikan bobot membuat algoritma bisa belajar dari kesalahan. Bila diprediksi harga selisih dengan harga asli masih terlalu besar, algoritma akan mengganti bobot sampai mencapai nilai yang akurat.

Setelah semua bobot diperbarui, proses feed forward dan backpropagasi diulang. Fase yang berulang disebut epoch. Oleh karena itu algoritma membutuhkan banyak waktu untuk mengulangi prosesnya. Setiap epoch berjalan, maka bobotnya akan lebih disesuaikan sehingga pada akhir epoch target bisa memberikan hasil prediksi yang akurat. Dalam penelitian ini digunakan target epoch sebanyak 10.000 kali.

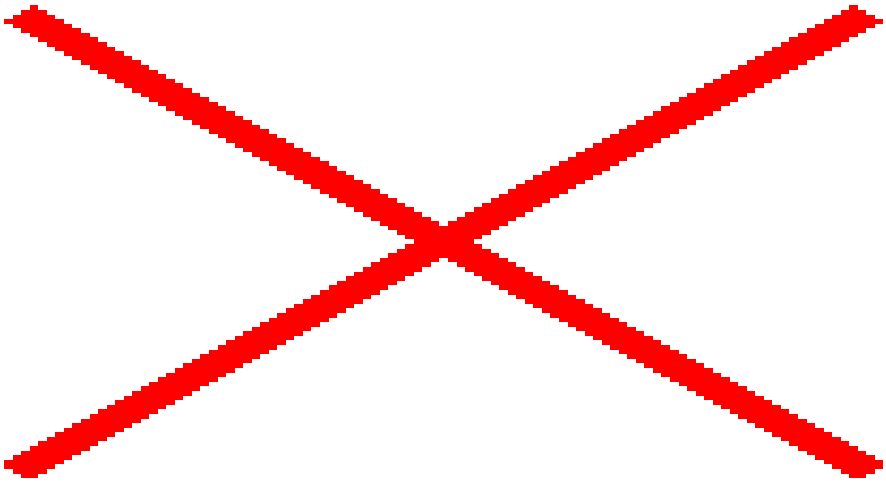
Hitung Prediksi

Hitung prediksi yang dilakukan setelah proses feed forward dan proses backpropagation berjalan sebanyak 10.000 epoch. Untuk menghitung prediksi diperlukan input harga terbuka, tinggi, rendah, menyesuaikan harga dan volume tanggal yang harus diprediksi. Data masukan dihitung dengan menggunakan tahap Feed Forward dan menggunakan bobot terakhir yang diperbarui. Hasilnya adalah prediksi harga yang mendekati.

Hasil Pengujian

Penelitian ini menggunakan data saham dari PT Telekomunikasi Indonesia (TLKM), PT HM Sampoerna (HMSP), PT Bank Central Asia (BBCA), PT Bank Rakyat Indonesia (BBRI) dan PT Astra International (ASII). Data pelatihan yang digunakan selama 1 tahun mulai Januari 2016 sampai desember 2016 (berjumlah sekitar 252 data). Pengujian data yang digunakan selama bulan januari 2017 - november 2017 (sekitar 241 data). Menggunakan epoch 10.000 dan learning rate 1.

Tabel 1. Hasil Pengujian



Kesimpulan

Penelitian ini bertujuan untuk mengetahui jumlah sel lapisan tersembunyi yang memberikan prediksi paling akurat. Komposisi jumlah sel lapisan tersembunyi adalah 2,

4, 6 dan 8 sel. Contoh kasus yang digunakan adalah memprediksi harga saham. Prediksi saham dilakukan dengan menggunakan data masa lalu

Hasil penelitian ini menemukan bahwa prediksi hasil semua jumlah sel memiliki akurasi di atas 99%. Setiap nomor sel pada lapisan tersembunyi memberikan hasil yang baik dalam menghitung prediksi harga saham.

Dalam penelitian ini disimpulkan bahwa variasi jumlah sel pada lapisan tersembunyi tidak mempengaruhi hasil prediksi. Perbedaan akurasi antara semua variasi jumlah sel pada hidden layer kurang dari 1% sehingga dianggap tidak berpengaruh.

Daftar Pustaka

[1] A. Triyono, A. J. Santoso, and Pranowo -, "Penerapan Metode Jaringan Syaraf Tiruan Backpropagation Untuk Meramalkan Harga Saham (IHSG)," *Jurnal Sistem dan Informatika (JSI)*, vol. 11, no. 1, Art. no. 1, 2016.

[2] S. Amiroch, "PREDIKSI HARGA SAHAM MENGGUNAKAN JARINGAN SYARAF TIRUAN BACKPROPAGATION," *Unisda Journal of Mathematics and Computer Science (UJMC)*, vol. 1, no. 01, Art. no. 01, Jun. 2015.

[3] T. Sutojo, *Kecerdasan buatan*. Andi Offset, 2011.

[4] R. B. Afrianto, H. Tjandrasa, and I. Arieshanti, "PREDIKSI PERGERAKAN HARGA SAHAM MENGGUNAKAN METODE BACK PROPAGATION NEURAL NETWORK," vol. 3, no. 3, p. 10, 2013.