

Penerapan Algoritma Genetika pada Optimalisasi Tim Pengerja Musik Gereja

Alwin Rengku^{#1}, Inge Martina^{#2}

*#Teknik Informatika, Institut Teknologi Harapan Bangsa
Jl. Dipatiukur no. 80-84, Bandung 40132, Indonesia*

¹winrengku@gmail.com

²inge@ithb.ac.id

Abstract— A church generally employs some music servants whom scheduled every week sequentially to some distributed locations. Each of them is divided into several different groups. They also have different level of expertise, so then the optimization solution should propose a desired team configuration. Every location has their own characteristic level, so it may require special team configuration. Genetic algorithm define servant as allele, service time slot in each location as gene, and servant allocation to each location as chromosome. Each chromosome proposed an alternative solution. Chromosome will be processed through some selection, crossover, and mutation steps, so then the best solution will be acquired. Best solution will be chosen from a chromosome that has fitness value near to 0, which means it is the most optimum solution. Population will be regenerated as long as the provided generation size. The regeneration process will be terminated if the best chromosome fitness does not change in a provided generation count. The average required generation to acquire solution from 200 servants in 9 locations is 12, with the crossover probability of 0.167 and mutation probability of 0.125.

Keywords— Optimization, church music servant team, genetic algorithm, regeneration termination, fitness chromosome value.

Abstrak— Gereja memiliki sejumlah pengerja musik yang dijadwalkan setiap pekan secara bergilir pada lokasi yang berbeda-beda. Pengerja terpisah kedalam beberapa kelompok, selain itu pengerja juga memiliki tingkat kemampuan yang berbeda-beda, sehingga solusi optimasi harus bisa menghasilkan susunan tim yang memenuhi ketentuan. Tidak hanya pengerja, setiap lokasi juga memiliki tingkatan karakteristik yang berbeda sehingga sebagian lokasi membutuhkan konfigurasi tim yang khusus. Algoritma genetika digunakan dengan menjadikan pengerja sebagai alel, waktu ibadah dalam setiap lokasi sebagai gen, dan alokasi pengerja pada setiap waktu ibadah dalam setiap lokasi sebagai kromosom. Setiap kromosom mewakili sebuah solusi. Kromosom akan melewati sejumlah tahapan seleksi, persilangan, dan mutasi, sehingga pada akhirnya dihasilkan sejumlah alternatif solusi terbaik. Solusi terbaik dipilih berdasarkan nilai *fitness* kromosom yang lebih mendekati 0, yang dalam hal ini berarti sangat optimal. Populasi akan mengalami regenerasi sejumlah ukuran generasi yang ditetapkan. Proses regenerasi akan berakhir jika *fitness* kromosom terbaik tidak mengalami perubahan selama jumlah generasi yang ditetapkan juga. Rata-rata generasi yang dibutuhkan untuk menghasilkan solusi dari 200 pengerja pada 9 lokasi adalah 12, dengan probabilitas persilangan 0,167 dan probabilitas mutasi 0,125.

Kata kunci— optimalisasi, tim pengerja musik gereja, algoritma genetika, terminasi regenerasi, nilai *fitness* kromosom.

I. PENDAHULUAN

Untuk dapat mengatasi permasalahan optimasi tim pengerja musik gereja dalam pengalokasiannya ke dalam setiap lokasi setiap minggunya diperlukan proses yang cukup kompleks. Algoritma genetika digunakan untuk mengoptimasi pengalokasian tim pengerja agar mendapatkan hasil yang terbaik. Dalam prosesnya terdapat beberapa masalah yang akan ditanggulangi.

Masalah-masalah tersebut meliputi adanya tiga *level* untuk pengerja dan untuk lokasi. *Level* tersebut adalah *beginner*, *intermediate*, dan *advanced*. Pengerja dengan *level beginner* tidak diprioritaskan untuk dialokasi ke dalam lokasi manapun kecuali dibutuhkan. Pengerja dengan *level intermediate* tidak diprioritaskan untuk dialokasi ke dalam lokasi dengan *level advanced* kecuali dibutuhkan. Bobot pelanggaran seorang pengerja bervariasi disesuaikan dengan keahlian dari pengerja tersebut. Perhitungannya dapat dilihat pada rumus 2.

Setiap lokasi memiliki dua waktu ibadah, yaitu pagi dan sore. Dalam suatu waktu ibadah terdapat jumlah ibadah yang bervariasi untuk setiap lokasi. Setiap lokasi pada umumnya akan diisi dengan tim pengerja yang tergabung berdasarkan beberapa keahlian tertentu yang kemudian akan dialokasikan pada waktu ibadah tertentu, tim tersebut sebisa mungkin berasal dari grup yang sama. Seluruh pengerja dipisah menjadi empat grup. Perhitungannya dapat dilihat pada rumus 3. Tim tersebut diwajibkan untuk melayani semua ibadah sepanjang waktu ibadah tersebut. Hal ini menjadi penyebab sulitnya menyamaratakan jumlah pelayanan setiap pengerja. Perhitungannya dapat dilihat pada rumus 4.

Selain itu setiap lokasi memiliki kapasitas jumlah pengerja yang bervariasi untuk setiap keahlian pengerja. Terdapat jumlah minimum dan maksimum untuk setiap keahlian dalam suatu lokasi. Dalam mengalokasi pengerja dengan keahlian tertentu, tujuan utamanya adalah untuk mencapai jumlah maksimum dari keahlian tersebut meskipun harus menggunakan *beginner*, namun jika tidak memungkinkan untuk mencapai jumlah maksimum maka digunakanlah jumlah minimum. Jumlah pengerja yang dialokasi tidak diperbolehkan melewati (kurang dari) jumlah minimum.

Setiap tim yang ada di lokasi tertentu sebisa mungkin grupnya bervariasi antar waktu ibadahnya. Perhitungannya dapat dilihat pada rumus 5. Selain itu untuk pengerja yang

melayani di bidang vokal sebisa mungkin tersusun dalam formasi yang seimbang (berjumlah ganjil) dan bervariasi *gender*-nya. Perhitungannya dapat dilihat pada rumus 6.

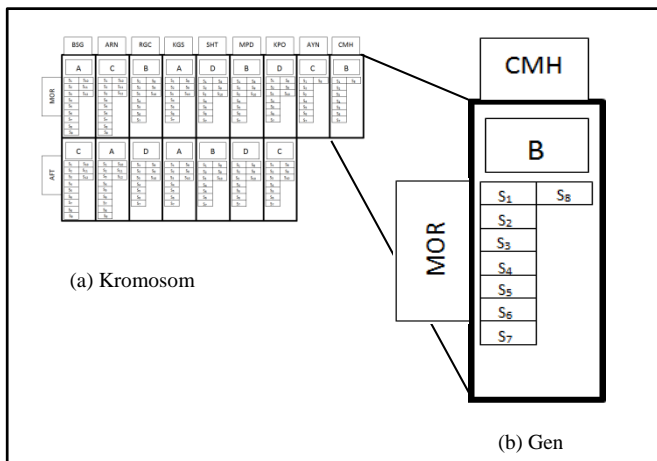
II. OPTIMASI TIM PENERJA MUSIK GEREJA

Pada Algoritma Genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Kromosom terdiri atas sejumlah gen, dan setiap gen merepresentasikan suatu entitas yang secara struktur berbeda dengan gen lainnya [1].

Setiap kromosom adalah sebuah individu yang akan merepresentasikan suatu solusi. Dalam penelitian ini individu/kromosom diimplementasikan dari suatu jadwal yang dibuat untuk suatu minggu tertentu. Kromosom akan terdiri dari sejumlah gen, dalam hal ini satu gen merupakan perwujudan dari satu waktu ibadah yang dibawakan oleh satu tim pengerja di suatu lokasi tertentu. Jumlah gen dalam satu kromosom didapat dari akumulasi jumlah waktu ibadah pada setiap lokasi.

Dalam ruang lingkup penelitian ini terdapat sembilan lokasi yang dikelompokkan menjadi tiga *level*, yakni *advanced* (AD), *intermediate* (IM), dan *beginner* (BG). Waktu ibadah dibagi menjadi pagi dan sore. Setiap waktu ibadah diisi dengan sejumlah ibadah, dan jumlahnya pun bervariasi untuk setiap lokasi.

Dan untuk masing-masing lokasi memiliki *skill slot* dengan jumlah beragam, untuk membatasi jumlah pengerja yang akan dialokasikan di lokasi tersebut. Semua pengerja berjumlah 200 dan disebar kedalam empat grup, yakni grup A, B, C, dan D. Tingkat keterampilan setiap pengerja dibagi dalam tiga *level*, yakni *advanced* (AD), *intermediate* (IM), dan *beginner* (BG). Berikut ini adalah model kromosom yang digunakan dalam proses optimasi.



Gambar 1 Model Kromosom dan Susunan Gennya

Kromosom di atas digambarkan dalam bentuk dua dimensi. Lokasi dijabarkan secara horizontal, dan waktu ibadah secara vertikal. Kromosom terdiri atas sejumlah gen. Pada gambar di atas gen menyimpan informasi mengenai lokasi (CMH),

waktu ibadah (MOR/Morning), grup (B) dan daftar pengerja ($S_1, S_2, S_3, \dots, S_n$) yang melayani di peribadatan tersebut. Waktu ibadah menyatakan apakah pagi atau sore, sedangkan grup menyatakan grup mana yang pengerjanya paling dominan dalam sesi tersebut. *Slot* S_1 sampai S_n akan diisi oleh masing-masing satu orang pengerja.

Pengalokasian *slot* dalam setiap gen mengacu pada jumlah maximum dan minimum *slot* pengerja dari lokasinya atas setiap *skill* yaitu *worship leader* (WL), *singer* (SG), *keyboardist* (KY), *bassist* (BS), *drummer* (DR), dan *guitarist* (GR). Pada gambar di atas pengalokasian *slot* digambarkan secara tergabung, misalkan S_1-S_7 (kolom kiri) merupakan akumulasi jumlah *slot* minimum, dan S_8 (kolom kanan) merupakan selisih jumlah *slot* menuju maksimum.

Fitness digunakan untuk menentukan kualitas suatu kromosom seiring dengan perubahannya dari generasi ke generasi oleh pindah silang dan mutasi. Setiap hasil perhitungan dari semua konstrain akan ditambahkan, kromosom dengan hasil perhitungan yang paling mendekati nol lah yang paling *fit*. Rumus *fitness* dari proses optimasi ini adalah:

$$F = F_1 + F_2 + F_3 + F_4 + F_5 \quad (1)$$

Rumus F_1 berfungsi untuk menghitung pelanggaran *level* yang terjadi dalam penempatan pengerja ke suatu lokasi tertentu. Pelanggaran terjadi apabila teralokasikan pengerja *level beginner* dalam suatu lokasi ataupun terdapat pengerja *level intermediate* di lokasi berlevel *advanced*. Selain itu terdapat beberapa *skill* yang memiliki bobot pelanggaran yang berbeda, yakni WL, musisi, dan *singer*. Bobot pelanggaran WL lebih besar dari musisi, dan bobot pelanggaran musisi lebih besar dari *singer*. Berikut ini adalah rumus F_1 :

$$F_1 = \sum_{i=1}^G \sum_j^H f(L_i, P_{ij}) \left(\frac{c}{P_{ij}} \right) \left(\frac{(L_i + 0,5)\beta_{p_{ij}} + \gamma_{p_{ij}} + 0,5}{\alpha_{p_{ij}}} \right)^{\alpha_{p_{ij}}} + (L_i + 0,5)\beta_{p_{ij}} + \gamma_{p_{ij}} \quad (2)$$

Tujuan dari rumus 2 adalah memberikan bobot perhitungan untuk setiap pengerja di dalam setiap gen. Setiap gen dan pengerja dinyatakan dengan indeks i , dan j . Jumlah gen dinyatakan dengan G dan H untuk jumlah pengerja dalam gen tersebut. Jika pengerja seorang WL bobot pelanggaran akan dikalikan dengan nilai $\alpha_{p_{ij}} = 1$, jika tidak maka $\alpha_{p_{ij}} = 0$. Jika pengerja seorang musisi bobot pelanggaran akan dikalikan dengan nilai $\beta_{p_{ij}} = 1$, jika tidak maka $\beta_{p_{ij}} = 0$. Jika pengerja seorang *singer* bobot pelanggaran akan diberikan nilai $\gamma_{p_{ij}} = 1$, jika tidak maka $\gamma_{p_{ij}} = 0$.

Bobot perhitungan kemudian akan dikalikan dengan sebuah pengali dengan perhitungan $f(L_i, P_{ij}) \times \frac{c}{P_{ij}}$. Dari pengali

tersebut akan ditentukan apakah pengerja tersebut melanggar dan seberapa besar pelanggarannya. Apabila pengerja tidak melanggar, perhitungan di atas akan memberikan nilai nol.

L_i adalah bobot *level* dari lokasi, P_{i_j} adalah bobot *level* dari pengerja, α_i adalah jumlah total WL, β_i adalah jumlah total musisi, γ_i adalah jumlah total *singer*.

Rumus F_2 berfungsi untuk menghitung tingkat keseragaman pengerja dalam suatu tim yang dibedakan berdasarkan grup. Dalam aturan ini tidak diperkenankan terdapat variasi grup antar pengerja di dalam suatu tim. Jika terjadi ketidakseragaman maka nilai *fitness* akan dihitung berdasarkan perhitungan terhadap dua divisi yakni divisi *singer* dan divisi musik. Perhitungan divisi musik memiliki bobot lebih besar dari pada divisi *singer*. Berikut ini adalah rumus F_2 :

$$F_2 = \sum_{i=1}^G \sum_{j=1}^D \frac{j^2}{2} C \left(\begin{array}{l} (R_{i_j} - 1) - \frac{\sum_{k=1}^{R_{i_j}} \left(1 - \frac{(P_{tot_{i_j}})_k}{P_{tot_{i_j}}} \right) \left(2 - \frac{(P_{tot_{i_j}})_k}{P_{tot_{i_j}}} \right)}{R_{i_j}} \\ + \left(\sum_{m=1}^{W_{i_j}} \left(1 - \frac{(P_{tot_{i_j}})_m}{P_{tot_{i_j}}} \right) \right) \end{array} \right) \quad (3)$$

Untuk setiap gen akan dilakukan perhitungan per divisi. Jumlah gen dinyatakan dengan G , dengan i sebagai indeks gen. Divisi terbagi dua yaitu divisi *singer* dan divisi musisi. Jumlah divisi dinyatakan dengan D , dengan j sebagai indeks divisi ($j = 1$ untuk *singer*, $j = 2$ untuk musisi). Jumlah variasi grup pengerja dalam suatu divisi dihitung dengan nilai R_{i_j} . Jika variasi grup lebih dari satu maka, untuk setiap kelompok grup akan diperhitungkan nilai pengaruhnya terhadap divisi tersebut dengan menggunakan persentase:

$$\frac{(P_{tot_{i_j}})_k}{P_{tot_{i_j}}}$$

Untuk penjelasannya adalah sebagai berikut:

$(P_{tot_{i_j}})_k$ = jumlah dari pengerja grup tertentu di suatu divisi.

$P_{tot_{i_j}}$ = total pengerja suatu divisi.

Namun dikarenakan dalam perhitungan *fitness* nilai terbaik adalah yang mendekati nol dan agar supaya akumulasi setiap nilai pengaruh memberikan hasil yang tepat, maka digunakanlah perhitungan berikut untuk menghitung nilai pengaruh dari setiap kelompok grup:

$$\left(1 - \frac{(P_{tot_{i_j}})_k}{P_{tot_{i_j}}} \right) \left(2 - \frac{(P_{tot_{i_j}})_k}{P_{tot_{i_j}}} \right)$$

Setelah diakumulasi semua nilai pengaruh dalam divisi tersebut, selanjutnya akan ditambahkan nilai pengaruh dari tingkat dominasi WL terhadap divisi tersebut dengan perhitungan:

$$\sum_{m=1}^{W_{i_j}} \left(1 - \frac{(P_{tot_{i_j}})_m}{P_{tot_{i_j}}} \right)$$

Tujuan dari perhitungan di atas adalah menghitung seberapa banyak pengerja dalam divisi tersebut yang memiliki kesamaan grup dengan (para) WL di gen tersebut kemudian ditentukan nilai pengaruhnya.

Rumus F_3 berfungsi untuk menghitung tingkat pemerataan jumlah pelayanan pengerja, dilihat dari berapa kali mereka melayani di suatu ibadah. Pengerja *beginner* tidak diikutsertakan dalam perhitungan ini karena tidak diprioritaskan untuk dialokasikan. Berikut ini adalah rumus F_3 :

$$F_3 = \sqrt{\frac{\sum_{i=1}^n (\beta_i - \bar{\beta})^2}{n}} \times C \quad (4)$$

Rumus F_3 merupakan implementasi dari rumus simpangan baku dalam menghitung tingkat pemerataan jumlah pelayanan seluruh pengerja. Nilai β merupakan selisih nilai antara jumlah pelayanan seorang pengerja dengan nilai rata-rata jumlah pelayanan dari seluruh pengerja. Nilai n merupakan jumlah seluruh pengerja. Dalam hal ini seluruh pengerja yang dihitung kemerataannya adalah hanya pengerja berlevel *advanced* dan *intermediate* saja.

Rumus F_4 berfungsi untuk menghitung terjadinya pengalokasian grup tim yang sama di seluruh waktu ibadah di setiap lokasi. Berikut ini adalah rumus F_4 :

$$F_4 = C\alpha^3 \quad (5)$$

Misalnya pada lokasi L1 dialokasikan tim pengerja grup A di waktu ibadah pagi dan sorenya, kondisi tersebut akan dihitung sebagai satu pelanggaran. Jumlah pelanggaran (α) kemudian akan dipangkatkan tiga.

Rumus F_5 berfungsi untuk menghitung ketidakseimbangan komposisi *gender* vocal (WL dan *singer*). Untuk melakukan perpaduan suara yang seimbang, maka perlu dilakukan penyeimbangan *gender* terhadap semua vokal yang ada dalam tim tersebut. *Fitness* akan bertambah buruk jika terdapat ketidakseimbangan *gender* misalnya dalam tim terdapat 6 pengerja vokal, 5 diantaranya perempuan hanya 1 laki-laki. Berikut ini adalah rumus F_5 :

$$F_5 = \sum_{i=1}^G (|a_i - b_i| - 1 + x) C \quad (6)$$

Pada perhitungan di atas akan dilakukan perhitungan untuk setiap gen. Simbol a menyatakan jumlah vokal wanita, sedangkan b menyatakan jumlah vokal pria. Nilai x berfungsi

sebagai *penalty* jika salah satu nilai baik itu *a* ataupun *b* ada yang bernilai nol, nilai *x* akan bernilai *a + b* jika tidak maka *x* bernilai nol.

Dalam perhitungan *fitness*, digunakan suatu nilai konstanta (*C*). Nilai konstanta ini didapat dari hasil bagi dari nilai perbandingan minimum dengan nilai pelanggaran minimum. Pertama-tama ditentukan nilai pelanggaran minimum dari setiap rumus yang ada. Setelah itu *user* akan melakukan penyesuaian dengan menentukan nilai perbandingan minimum terhadap setiap nilai pelanggaran minimum sehingga bobot pengaruh untuk setiap *constrain* dapat diatur sesuai kebutuhan. Tabel I memuat tetapan konstanta (*C*) berdasarkan nilai minimum dari tiap perhitungan yang ada.

III. PENGUJIAN

Pengujian pada penelitian ini dilakukan untuk mengetahui tingkat kinerja dari algoritma genetika yang telah dibuat dengan menggunakan *crossover probability* dan *mutation probability* [2]. Berikut ini adalah kriteria dari pengujian:

- *Crossover Probability* (*Pc*) = 0,167, *Mutation Probability* (*Pm*) = 0,125
- *Crossover Probability* (*Pc*) = 1, *Mutation Probability* (*Pm*) = 1

Terdapat hasil akhir *output* yang akan dibandingkan adalah sebagai berikut:

- *Best Fitness* (BF): merupakan *fitness* dari solusi terbaik sekaligus sebagai hasil optimasi.
- *Best Fitness Difference Amount* (BFDA): artinya selisih dari perubahan *fitness* terbaik antara generasi awal dengan generasi terakhir (dihitung berdasarkan *fitness*).
- *Best Fitness Alteration Amount* (BFAA): artinya jumlah perubahan *best fitness* yang terjadi dari generasi awal sampai generasi akhir (dihitung berdasarkan iterasi).
- *Generation Reached* (GR): artinya jumlah generasi yang dilampui hingga mendapatkan solusi terbaik.

Selanjutnya empat nilai tersebut akan digunakan untuk menentukan nilai *Genetic Algorithm Performance Rate* (GAPR). Perhitungan ini dirancang oleh penulis hanya sebagai perbandingan untuk menilai kinerja dari Algoritma Genetika yang diterapkan dalam sebuah pengujian. Nilai GAPR ditentukan menggunakan rumus berikut ini.

$$GAPR = \frac{BFDA \times BFAA}{(BF \times GR) + 1} \quad (7)$$

Dilakukan masing-masing sepuluh kali terhadap setiap kriteria pengujian. Ringkasan dari hasil pengujian tersebut adalah sebagai berikut.

- $Pc = 0,167, Pm = 0,125 \cdot \overline{GAPR} = 1,14 \cdot 10^{-2}$
- $Pc = 1, Pm = 1 \cdot \overline{GAPR} = 9,66 \cdot 10^{-4}$

Berdasarkan hasil pengujian di atas dapat disimpulkan bahwa semakin kecil nilai *Pc*, dan *Pm* akan memberikan pengaruh yang semakin baik terhadap kinerja Algoritma Genetika.

TABEL I
PENENTUAN BOBOT KONSTANTA

Rumus	Nilai Perbandingan Minimum (u)	Nilai Pelanggaran Minimum (m)	C (u/m)
F ₁	2,5	0,5	5
F ₂	2	0,153	13,07
F ₃	0,8	0,079	10,12
F ₄	20	1	20
F ₅	0,5	1	0,5

Dari kedua kriteria pengujian yang bisa memberikan kinerja terbaik adalah kriteria pertama dengan rata-rata GAPR sebesar 0,0114 dari sepuluh pengujian. Kriteria pertama memiliki *Pc*, dan *Pm* yang kecil, yang hanya menukarkan satu pasang pengerja saja. Hal ini memberikan keuntungan yakni jika pengerja yang ditukarkan tepat maka perubahan *fitness* dapat langsung terjadi meskipun hanya sedikit. Kriteria kedua mendapatkan hasil yang lebih buruk karena dari sepuluh pengujian sangat jarang ada yang berhasil mengubah *fitness* terbaik dari generasi pertama. Hal ini mungkin terjadi karena *Pc* dan *Pm* bernilai satu. Artinya suatu kromosom dalam upaya pindah silangnya menukarkan enam pasang pengerja sekaligus, begitu pula dengan mutasi, sehingga perbaikan nilai *fitness* tidak bisa dikontrol. Misalnya dua pasang pengerja diantaranya menghasilkan hasil pertukaran dengan nilai *fitness* berkurang 10 dari kromosom terbaik, namun pertukaran empat pasang yang lain menghasilkan hasil pertukaran dengan nilai *fitness* bertambah 20, sehingga bukan malah membaik sebaliknya nilai *fitness* dari kromosom tersebut memburuk.

IV. KESIMPULAN

Kesimpulan yang didapat oleh penulis dari pembuatan aplikasi ini yaitu: Algoritma Genetika dapat diterapkan pada optimasi tim pengerja musik pengerja dalam skala mingguan dengan hasil yang optimal. Selain itu dalam mengolah data yang banyak, proses pindah silang dan mutasi merupakan alternatif yang sangat diperlukan kinerja terbaiknya, dalam proses evolusi terhadap populasi.

DAFTAR REFERENSI

- [1] Sivanandam, Deepa. *Introduction to Genetic Algorithms: Evolutionary Algorithm*. New York: Springer Berlin Heidelberg, 2008.
- [2] Torres Jairo, Franco Edgar, Mayorga Carolina. 2010. "Project Scheduling with Limited Resources Using A Genetic Algorithm". *International Journal of Project Management*, vol. 28, pp. 619–628.

Alwin Rengku, menempuh pendidikan S1 di Program Studi Teknik Informatika ITHB Bandung dan memperoleh gelar Sarjana Teknik pada tahun 2015.

Inge Martina, menyelesaikan S1 di Jurusan Teknik Informatika ITB pada Maret 1990 dan melanjutkan S2 di Jurusan Teknik Informatika ITB hingga selesai pada bulan Agustus 2014. Minat penelitian pada efisiensi algoritma.