

Algoritma Pencarian untuk Pemetaan Kecocokan pada Program Pertukaran Ginjal

Fazat Nur Azizah
fazat@informatika.org

Ardian Umam
ardian@informatika.org

Leonardo
leonardow41@gmail.com

Abstrak—Sebelum transplantasi ginjal dilakukan, pendonor dan pasien calon penerima ginjal harus merupakan pasangan yang cocok. Faktanya, tidak jarang ketidakcocokan terjadi di antara pasangan donor-resipien, menimbulkan masalah ketersediaan ginjal. Untuk menyelesaikan masalah ketersediaan ini, Program Pertukaran Ginjal (*Kidney Exchange program*) diciptakan agar pasangan-pasangan ketidakcocokan dapat melakukan pertukaran ginjal dengan pasangan ketidakcocokan lainnya, membuat pendonoran silang. Namun karena sangat banyaknya pasangan ketidakcocokan, mencari pemetaan kecocokan dari sekumpulan pasangan ketidakcocokan bukanlah hal yang mudah. Oleh karena itu, algoritma pencarian pemetaan kecocokan dibuat. Algoritma yang sudah ada, *Edmond's Algorithm*, tidaklah optimal karena sifatnya yang menutup kemungkinan terjadinya pertukaran tiga arah atau lebih. Pada paper ini, dengan modifikasi algoritma pencarian pemetaan kecocokan dua arah yang sudah ada, diimplementasikan algoritma pencarian pemetaan kecocokan N arah. Berdasarkan hasil pengujian, menggunakan data berisikan 3000 pasangan ketidakcocokan, terbukti bahwa algoritma N arah adalah algoritma yang superior dibandingkan dengan algoritma dua arah apabila diukur menggunakan metrik efisiensi pencocokan (*matching efficiency*). Peningkatan efisiensi pencocokan yang didapatkan secara rata-rata mencapai 7.75% jika dibandingkan dengan efisiensi pencocokan yang didapatkan oleh *Edmond's Algorithm*.

Kata Kunci—transplantasi ginjal, pertukaran ginjal, algoritma pencarian pemetaan kecocokan

I. PENDAHULUAN

Transplantasi ginjal merupakan metode perawatan yang paling direkomendasikan untuk penyakit-penyakit ginjal yang serius [1]. Setiap tahunnya, di Indonesia sendiri, terdapat lebih dari 100,000 pasien yang memiliki kebutuhan ginjal transplan. Namun hanya sekitar 20% dari pasien-pasien tersebut yang bisa mendapatkan transplantasi [2]. Hal ini terjadi karena adanya masalah finansial, pandangan masyarakat, dan yang paling umum, masalah ketersediaan. Meskipun jumlah pendonor setiap tahunnya semakin meningkat [3], masalah ketersediaan ginjal tetaplah ada dikarenakan oleh banyaknya kriteria medis yang harus dipenuhi oleh pasangan donor-resipien sebelum operasi dapat dilakukan [2]. Pendonor harus memiliki ginjal yang sehat, golongan darah yang cocok dengan resipien, dan tidak memiliki penyakit yang dapat menular melalui darah. Selain itu, sistem imun dari pasien juga harus dapat menerima ginjal pendonor tanpa membunuh ginjal yang bersangkutan.

A. Transplantasi Ginjal

Terdapat beberapa buah pengujian yang harus dilalui sebelum transplantasi ginjal dapat dilakukan [4]. Uji golongan

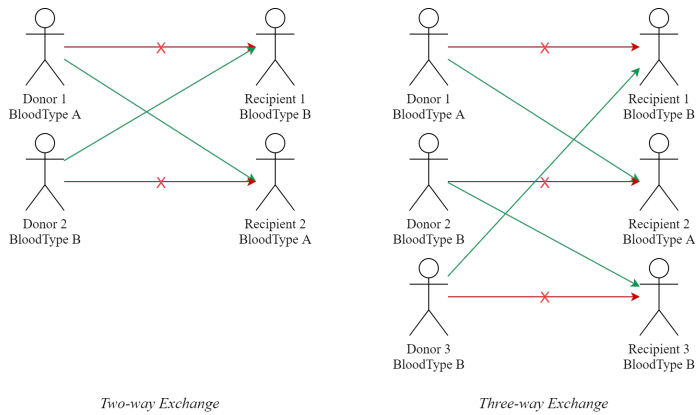
Tabel I
KOMPATIBILITAS GOLONGAN DARAH PENERIMA DAN PEMBERI DONOR GINJAL [5]

Resipien \ Donor	O	A	B	AB
O	1	0	0	0
A	1	1	0	0
B	1	0	1	0
AB	1	1	1	1

darah dilakukan untuk donor dan juga resipien. Seorang pendonor hanya dapat melakukan pendonoran apabila golongan darah pendonor bersifat cocok dengan golongan darah resipien. Seperti yang dapat dilihat pada Tabel I, angka 1 mengindikasikan kecocokan dan angka 0 mengindikasikan ketidakcocokan. Pendonor dengan golongan darah O merupakan *Universal Donor* karena dapat melakukan donor ke semua golongan darah. Resipien dengan golongan darah AB merupakan *Universal Recipient* karena dapat menerima donor dari semua golongan darah [6]. Uji imunitas dilakukan setelah pasangan donor-resipien melewati uji golongan darah [4]. Terdapat dua jenis pengujian yang dilakukan, yaitu uji silang dan uji *Human Leucocyte Antigen* (HLA). Pada uji silang, darah pendonor dipertemukan dengan darah resipien untuk melihat terjadinya penggumpalan atau tidak. Penggumpalan terjadi apabila sistem imun dari resipien menganggap darah dari pendonor merupakan benda asing yang harus dieksterminasi [7]. Uji HLA juga dilakukan untuk menguji perlawanan dari sistem imun pasien terhadap ginjal pendonor, namun digunakan sel tisu dari pendonor dan resipien melainkan darah [8]. Setelah semua pengujian selesai, uji serologi dilakukan pada pendonor untuk menguji apakah terdapat penyakit yang dapat menular melalui darah pada pendonor [7].

B. Program Pertukaran Ginjal

Karena banyaknya pasangan donor-resipien yang tidak cocok akibat prasyarat-prasyarat transplantasi, program pertukaran ginjal bernama *Kidney Paired Donation* (KPD) diciptakan agar pasangan donor-resipien ketidakcocokan dapat mempertukarkan ginjal dengan pasangan ketidakcocokan lainnya, membuat pendonoran silang menjadi mungkin [5]. Per-



Gambar 1. Pertukaran dua arah(kiri) dan tiga arah(kanan) pada Program Pertukaran Ginjal

tukaran ginjal antar-pasangan dapat dilakukan secara dua arah, tiga arah, dan dapat dilakukan hingga N arah.

Karena banyaknya pasangan ketidakcocokan, rumah sakit-rumah sakit tidak dapat secara mudah mendapatkan solusi pencocokan yang optimal untuk sekumpulan pasangan-pasangan ketidakcocokan. Dibutuhkan algoritma untuk mencari pemetaan kecocokan. Salah satu algoritma yang diketahui untuk menyelesaikan permasalahan ini adalah *Edmond's Algorithm* [5]. Pada *Edmond's Algorithm*, kumpulan pasangan ketidakcocokan direpresentasikan dengan suatu graf dengan setiap simpul yang adalah pasangan ketidakcocokan dan sisi yang merupakan arah pencocokan antar-pasangan yang mungkin. Algoritma ini berfokus untuk mencari kecocokan terlebih dahulu untuk pasangan-pasangan dengan prioritas tinggi sehingga pasien-pasien yang lebih membutuhkan transplantasi dapat mendapatkan pertukaran terlebih dahulu. Algoritma lain pada lingkup kerja ini adalah algoritma *First Accept Heuristic Match* [5] yang menggunakan heuristik dimana pasangan yang registrasi lebih dulu akan mendapatkan kecocokan terlebih dahulu.

Algoritma-algoritma ini dinamakan algoritma pencarian pemetaan kecocokan. Bagus atau tidaknya algoritma-algoritma ini diukur menggunakan metrik performa Efisiensi Pencocokan dan Waktu Eksekusi Algoritma [9]. Efisiensi pencocokan merepresentasikan berapa banyaknya pasangan yang mendapatkan kecocokan dari semua pasangan yang ada, ditulis dalam bentuk persentase. Waktu Eksekusi Algoritma adalah metrik yang mengukur seberapa lama waktu jalannya algoritma dimulai dari *input* masuk hingga *output* keluar, diukur dalam milisekon (ms). Algoritma pencarian pemetaan kecocokan terbaik merupakan algoritma yang mampu menghasilkan efisiensi pencocokan yang tinggi dalam waktu eksekusi yang singkat, yang berarti lebih banyak pasien yang lebih dapat terselamatkan secepat mungkin.

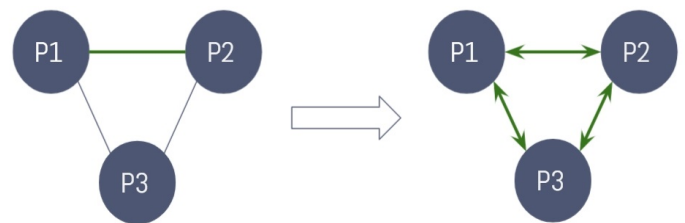
Meskipun *Edmond's Algorithm* dan *First Accept Heuristic Match* sudah ada, algoritma-algoritma tersebut hanya dapat mengembalikan pertukaran dua arah. Hal ini merupakan suatu batasan karena algoritma-algoritma ini menutup kemungkinan dicarikkannya solusi dengan pertukaran tiga arah atau lebih.

II. METODOLOGI

Berdasarkan masalah-masalah dari algoritma yang sudah dijelaskan pada bab sebelumnya, dibutuhkan algoritma pencarian pemetaan kecocokan N arah untuk kumpulan pasangan ketidakcocokan. Dengan memodifikasi algoritma-algoritma yang ada, algoritma-algoritma baru dapat dibuat.

A. Modifikasi-modifikasi untuk Algoritma yang sudah ada

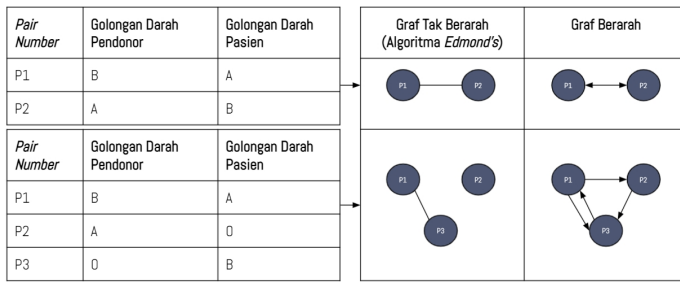
Sebelum algoritma baru dibuat, struktur data dari kumpulan pasangan ketidakcocokan perlu didefinisikan ulang. Graf Kompatibilitas yang digunakan pada *Edmond's Algorithm* adalah graf tak berarah [5], yang berarti sisi pada graf merepresentasikan kecocokan antara dua simpul, sehingga untuk mencari kecocokan antar pasangan, algoritma hanya perlu mendeteksi sisi pada graf kompatibilitas yang sudah ada. Hal ini menutup kemungkinan terdeteksinya kecocokan tiga arah atau lebih. Apabila dilakukan modifikasi untuk digunakan algoritma deteksi siklus dan bukan deteksi sisi untuk menggantikan ini, algoritma-algoritma yang sudah ada akan mampu mendeteksi kecocokan N arah. Seperti yang dapat dilihat pada Gambar 2, dengan data yang sama, dengan menggantikan algoritma deteksi sisi dengan deteksi siklus, kecocokan tiga arah dapat terdeteksi sehingga efisiensi pencocokan yang didapat meningkat, membuahkan hasil yang lebih baik.



Gambar 2. Deteksi Sisi(kiri) vs. Deteksi Siklus(kanan) pada Graf Kompatibilitas

Meskipun dengan mengubah deteksi sisi menjadi deteksi siklus berpotensi meningkatkan efisiensi pencocokan, hal ini tentunya tidak cukup. Perlu diperhatikan bahwa siklus-siklus yang didapatkan dari graf adalah siklus bolak-balik. Apabila graf kompatibilitas yang merupakan graf tak berarah dapat diubah menjadi graf berarah, siklus-siklus satu arah pada graf juga dapat dideteksi dan dikembalikan, yang secara tidak langsung menambahkan jumlah graf yang dikembalikan, sehingga berpotensi meningkatkan efisiensi pencocokan seperti yang dapat dilihat pada gambar 3. Graf tak berarah sendiri adalah struktur data yang terdiri dari sekumpulan simpul dan dihubungkan oleh sisi, dimana setiap sisi memiliki arah yang menunjukkan arah pergerakan yang dapat dilakukan dari suatu simpul ke simpul-simpul lainnya [10]. Algoritma deteksi siklus untuk graf berarah digunakan untuk mencari keberadaan jalur satu arah dari suatu simpul yang kembali ke simpul mulai pada graf [11].

Algoritma deteksi siklus yang digunakan berbeda dengan algoritma deteksi siklus pada umumnya yang mendeteksi apakah suatu graf memiliki siklus atau tidak (True/False question). Pada implementasi ini, algoritma akan mengumpulkan



Gambar 3. Graf Kompatibilitas Tak Berarah vs. Berarah

dan mengembalikan semua siklus yang dapat terbentuk dari suatu graf kompatibilitas. Setiap siklus pada list siklus yang dikembalikan adalah pertukaran ginjal yang mungkin terjadi karena bersifat cocok antar-pasangan. Panjang dari siklus merepresentasikan N pada pertukaran N arah (cth. siklus sepanjang 3 merepresentasikan pertukaran tiga arah). Modifikasi ini dilakukan dengan harapan jumlah siklus yang didapatkan akan meningkat niscaya meningkatkan efisiensi pencocokan, agar lebih banyak pasien yang dapat terselamatkan.

B. Algoritma Pencarian Pemetaan Kecocokan N arah

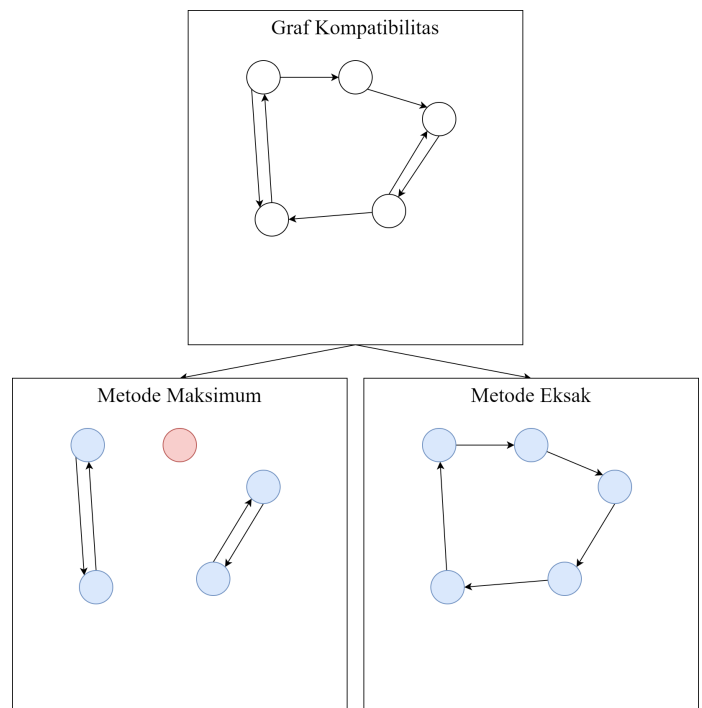
Dengan kompatibilitas graf yang berarah dan juga algoritma deteksi siklus, algoritma pencarian pemetaan kecocokan N arah dapat dibuat. Algoritma-algoritma N arah ini akan menerima list siklus yang dikembalikan dari algoritma deteksi siklus yang telah dijelaskan sebelumnya. Keluaran dari algoritma ini adalah list siklus yang telah tereduksi yang dinamakan sebagai pemetaan kecocokan. Reduksi yang terjadi pada algoritma dilakukan karena setiap pasangan pada graf kompatibilitas ada pada lebih dari satu siklus pada list siklus masukan. Hal ini tentunya tidak mungkin karena setiap pasangan hanya dapat memberikan dan menerima tepat satu buah ginjal. Reduksi ini dilakukan untuk mengubah hubungan *many to many* ini menjadi hubungan *one to one*. Reduksi ini pada umumnya menyebabkan beberapa pasangan ketidakcocokan untuk tidak mendapatkan kecocokan sama sekali. Oleh karena itu, metrik efisiensi pencocokan digunakan untuk menilai kualitas algoritma dalam menghasilkan jumlah pasangan yang mendapatkan kecocokan sebanyak-banyaknya dari kumpulan pasangan ketidakcocokan.

1) *First Accept Searching Algorithm*: Algoritma N arah yang pertama adalah algoritma *First Accept Searching*. Inspirasi dari algoritma ini adalah algoritma *First Accept Heuristic Match*. Pendekatan yang digunakan disini adalah untuk mengembalikan siklus-siklus yang ada terlebih dahulu. Algoritma ini memastikan tidak ada pasangan yang mendapatkan kecocokan lebih dari satu kali dengan menghilangkan siklus-siklus yang mengandung pasangan-pasangan yang telah ada pada siklus-siklus yang telah dikembalikan.

Algoritma ini menggunakan dua buah parameter. Parameter pertama tentunya adalah nilai N ini sendiri, karena algoritma ini adalah algoritma pencarian pemetaan kecocokan N arah. Parameter kedua adalah metode penggunaan nilai N . Nilai N dapat digunakan sebagai panjang kecocokan maksimum dan

sebagai panjang kecocokan eksak. Jika metode maksimum digunakan, maka siklus-siklus yang dikembalikan dapat memiliki panjang 2 hingga N . Sementara jika metode eksak digunakan, maka siklus-siklus yang dikembalikan hanya adalah siklus-siklus dengan panjang N saja.

Meskipun metode maksimum terlihat lebih unggul dibandingkan metode eksak, tidak selalu terjadi demikian. Sebagai contoh, seperti yang dapat dilihat pada Gambar 4, saat N adalah 5, saat metode maksimum digunakan, siklus-siklus bolak-balik pada graf akan dikembalikan, sehingga pertukaran lima arah pada graf akan dibuang. Saat metode eksak digunakan, karena algoritma akan mencari siklus-siklus lima arah, algoritma akan mengembalikan pertukaran lima arah pada graf. Pada kasus ini, efisiensi pencocokan yang lebih tinggi diperoleh saat metode eksak digunakan, bukan metode maksimum.



Gambar 4. Kasus dimana Metode Eksak unggul dibandingkan Metode Maksimum

2) *Priority-based Searching Algorithm*: Algoritma N arah yang kedua adalah algoritma *Priority-based Searching*. Inspirasi dari algoritma ini adalah *Edmond's Algorithm*. Cara kerja dari algoritma ini sangat mirip dengan algoritma *First Accept Searching*, namun setiap siklus diberikan nilai prioritas sehingga siklus-siklus yang memiliki prioritas tinggi dapat dikembalikan terlebih dahulu dibandingkan yang lainnya.

Algoritma ini menggunakan tiga parameter. Dua parameter pertama juga digunakan pada algoritma *First Accept Searching*, yaitu nilai N dan juga metode penggunaan nilai N . Parameter ketiga adalah metode penentuan prioritas. Terdapat dua metode untuk menentukan prioritas setiap siklus. Metode pertama adalah metode *greedy*. Metode ini memberikan prioritas yang lebih tinggi pada siklus-siklus yang

lebih panjang, dengan harapan dengan banyaknya siklus-siklus panjang yang dikembalikan, akan meningkatkan efisiensi pencocokan. Metode kedua adalah metode *infrequent*. Metode ini memberikan prioritas tinggi kepada siklus-siklus dengan pasangan-pasangan yang sulit mendapatkan kecocokan. Pada implementasinya, pasangan-pasangan ini diindikasikan sebagai suatu simpul yang terhubung oleh jumlah sisi yang sedikit, sehingga pasangan-pasangan tersebut lebih terbatas dari pasangan lainnya untuk mendapatkan kecocokan dengan pasangan-pasangan ketidakcocokan lainnya.

III. EXPERIMEN DAN ANALISIS

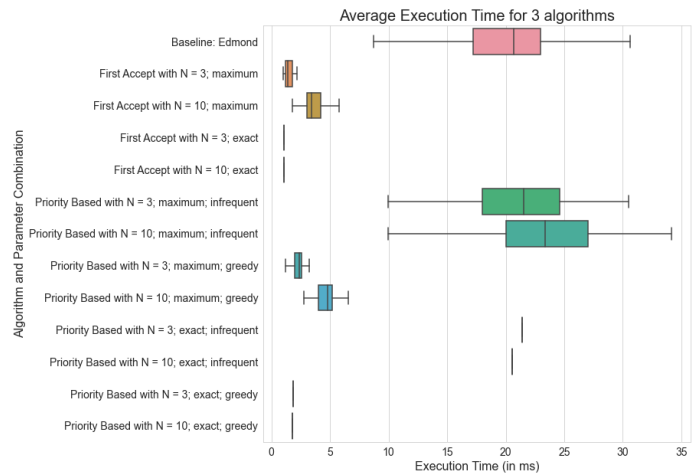
Sebelum eksperimen dilakukan, *Edmond's Algorithm* diimplementasikan sebagai algoritma *baseline* agar perbandingan performa antara algoritma *N* arah dan algoritma yang sudah ada dapat dilakukan. Data yang digunakan pada eksperimen ini dapat ditemukan pada pranala <https://rdm.inesctec.pt/dataset/ii-2020-001>. Kode sumber lengkap dari implementasi dan eksperimen dapat ditemukan pada pranala <https://github.com/Mingtaros/Kidney-Exchange-Match-Mapping-System>. Tujuan dari eksperimen dan evaluasi ini adalah untuk mencari kombinasi algoritma dengan parameter seperti apa kah yang mampu menghasilkan performa terbaik berdasarkan metrik-metrik yang telah ditentukan. Melalui eksperimen-eksperimen ini, tentunya juga diuji apakah algoritma *N* arah mampu mengungguli performa dari algoritma *baseline* yang digunakan, *Edmond's Algorithm*. Seperti yang telah dijelaskan, pengukuran performa algoritma dilakukan dengan metrik efisiensi pencocokan dan juga waktu eksekusi algoritma.

A. Perbandingan Waktu Eksekusi

1) *Perbandingan Waktu Eksekusi untuk Kombinasi Algoritma Berbeda*: Pada eksperimen ini, setiap algoritma akan memproses data berisi 3000 pasangan ketidakcocokan yang telah dipecah-pecah menjadi 30 grup yang setiap grupnya dinamakan sebagai suatu *seed*. Setiap *seed* berisikan 100 pasangan ketidakcocokan. Setiap algoritma akan dijalankan dengan masing-masing data *seed* 50 kali. Seperti yang dapat dilihat pada gambar 5, setiap poin data pada *boxplot* adalah waktu eksekusi rata-rata dari 50 kali eksekusi untuk setiap data *seed*. Untuk setiap koordinat Y, terdapat 30 titik data yang masing-masing merepresentasikan waktu eksekusi algoritma untuk *seed* yang bersangkutan.

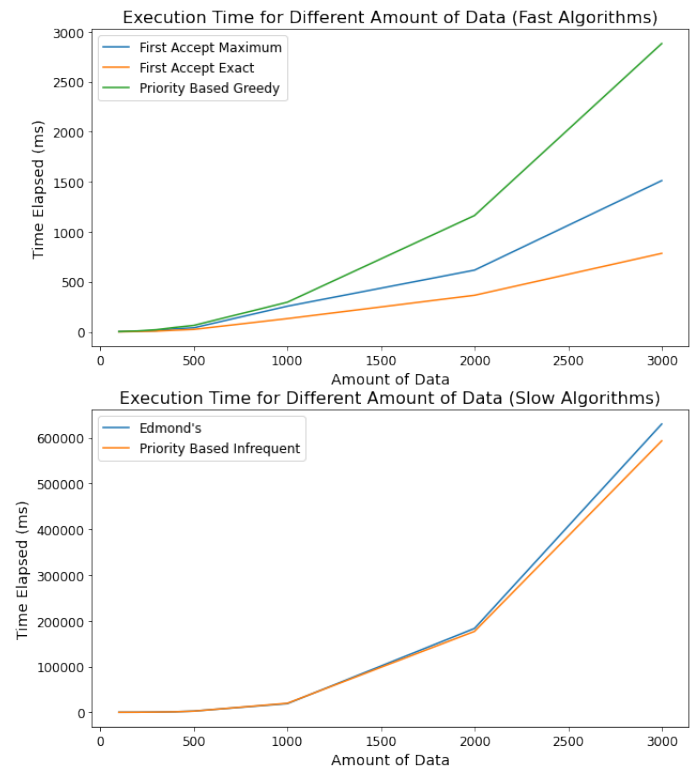
Seperti yang dapat dilihat pada *boxplot*, *Edmond's Algorithm* dan *Priority-based Searching* dengan metode penentuan prioritas *infrequent* adalah algoritma-algoritma yang paling lambat. Namun, seperti yang dapat dilihat, karena tidak ada perbedaan waktu eksekusi yang sangat signifikan antar algoritma, dapat disimpulkan secara kasar bahwa perbedaan waktu eksekusi antar algoritma dapat diabaikan.

2) *Perbedaan Waktu Eksekusi untuk Ukuran Data Berbeda*: Eksperimen ini dilakukan untuk melihat pengaruh dari ukuran data yang berbeda terhadap waktu eksekusi dari algoritma pencarian pemetaan kecocokan. Untuk eksperimen ini, dicatat rata-rata waktu eksekusi setiap algoritma untuk data sebesar 100, 200, 300, 400, 500, 1000, 2000, dan 3000 pasangan



Gambar 5. Rata-rata Waktu Eksekusi untuk ketiga Algoritma

ketidakcocokan. Setiap titik data pada Gambar 6 menandakan waktu eksekusi algoritma untuk jumlah data yang masing-masingnya dilakukan sebanyak 10 kali.



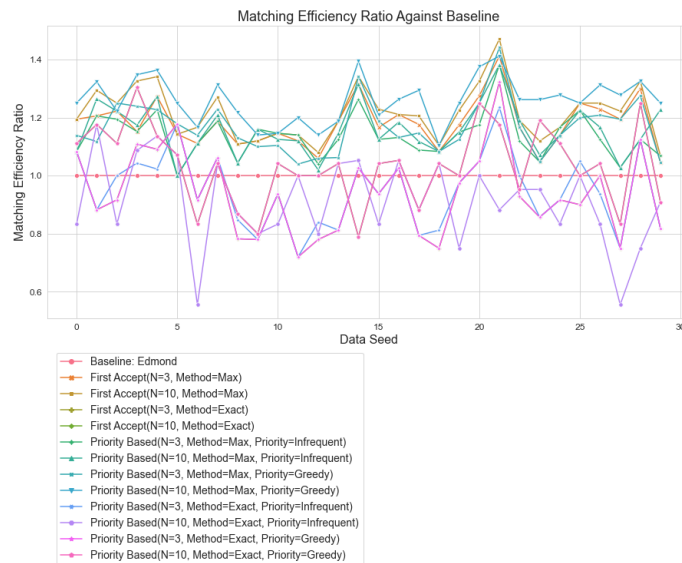
Gambar 6. Rata-rata Waktu Eksekusi untuk Ukuran Data Berbeda

Seperti yang dapat dilihat pada Gambar 6, terdapat dua buah diagram yang masing-masing merepresentasikan grup algoritma yang cepat dan grup algoritma yang lambat. Untuk algoritma yang cepat, saat digunakan 3000 pasangan, waktu maksimum dari eksekusi algoritma berkisar di 3000 milisekon (30 detik). Sementara untuk algoritma yang lambat, saat digunakan 3000 pasangan, waktu maksimum dari eksekusi

algoritma berkisar di 600000 milisekon (10 menit). Hal ini terjadi karena metode penentuan prioritas yang mirip pada *Edmond's Algorithm* dan juga metode penentuan prioritas *infrequent* pada algoritma *Priority-based Searching*. Dari kedua diagram, dapat disimpulkan bahwa relasi antara ukuran data dan waktu eksekusi berbentuk polinomial dan bukan linear.

B. Perbandingan Efisiensi Pencocokan secara menyeluruh untuk Algoritma Berbeda

Eksperimen terakhir dilakukan untuk menunjukkan kombinasi algoritma dan parameter mana yang adalah kombinasi terbaik untuk menghasilkan pemetaan kecocokan dengan efisiensi pencocokan yang tinggi. Pada eksperimen ini, setiap kombinasi diujikan menggunakan 30 data *seed* berisikan 100 data untuk masing-masing *seed*. Untuk mempermudah pengertian, pada Gambar 7, rasio perkembangan digunakan untuk menunjukkan besarnya efisiensi pencocokan dari suatu kombinasi algoritma parameter dibandingkan dengan *baseline* yang merupakan garis lurus horizontal pada gambar. Setiap titik data merepresentasikan rasio efisiensi pencocokan dari algoritma saat digunakan pada data *seed* tertentu dibandingkan dengan efisiensi pencocokan yang didapatkan oleh *Edmond's Algorithm*.



Gambar 7. Rasio Efisiensi Pencocokan dari Algoritma N arah dibandingkan Baseline

Untuk menyederhanakan, digunakan tabel II yang berisikan rata-rata rasio efisiensi pencocokan dibandingkan baseline. Rata-rata yang digunakan pada tabel adalah rata-rata rasio untuk setiap data *seed*.

Seperti yang dapat terlihat pada tabel II, mayoritas dari algoritma N arah menghasilkan efisiensi pencocokan yang lebih tinggi dibandingkan dengan *baseline*. Namun, untuk beberapa algoritma yang menggunakan metode penggunaan nilai N eksak, efisiensi pencocokan yang didapatkan lebih buruk dibandingkan jika digunakan *Edmond's Algorithm*. Secara

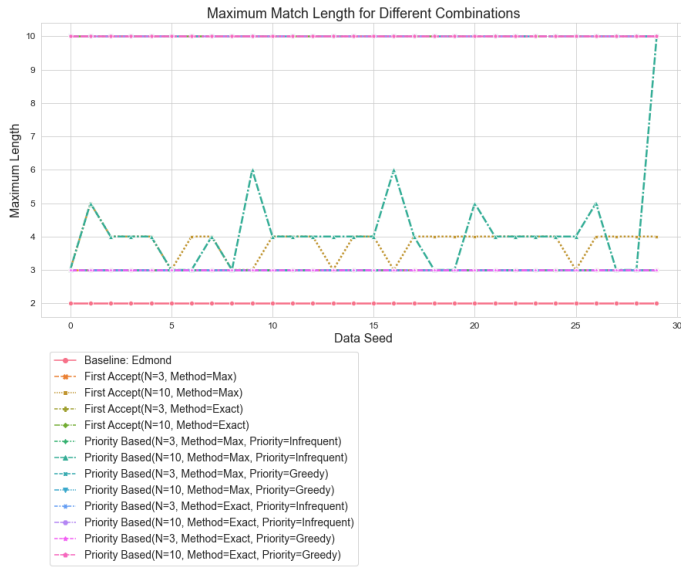
Tabel II
RATA-RATA RASIO EFISIENSI PENCOCOKAN DARI ALGORITMA N ARAH
DIBANDINGKAN BASELINE

Algoritma	Rata-rata Rasio Perkembangan
Baseline: Edmond's Algorithm	1.0
First Accept(N=3, Method=Max)	1.18
First Accept(N=10, Method=Max)	1.22
First Accept(N=3, Method=Exact)	0.94
First Accept(N=10, Method=Exact)	1.04
Priority Based(N=3, Method=Max, Priority=Greedy)	1.17
Priority Based(N=10, Method=Max, Priority=Greedy)	1.26
Priority Based(N=3, Method=Max, Priority=Infrequent)	1.14
Priority Based(N=10, Method=Max, Priority=Infrequent)	1.16
Priority Based(N=3, Method=Exact, Priority=Greedy)	0.94
Priority Based(N=10, Method=Exact, Priority=Greedy)	1.04
Priority Based(N=3, Method=Exact, Priority=Infrequent)	0.95
Priority Based(N=10, Method=Exact, Priority=Infrequent)	0.91

menyeluruh, rata-rata efisiensi pencocokan dari seluruh algoritma N arah lebih tinggi 7.75% dibandingkan jika digunakan *Edmond's Algorithm*. Peningkatan efisiensi pencocokan yang paling tinggi dibandingkan *baseline* adalah saat algoritma *First Accept Searching* digunakan dengan nilai N 10, metode penggunaan nilai N maksimum, pada data *seed* 10, dimana rasio efisiensi pencocokan sebesar 1.4706 dari *baseline*, mengartikan 47.06% peningkatan efisiensi pencocokan dibandingkan saat digunakan *Edmond's Algorithm*.

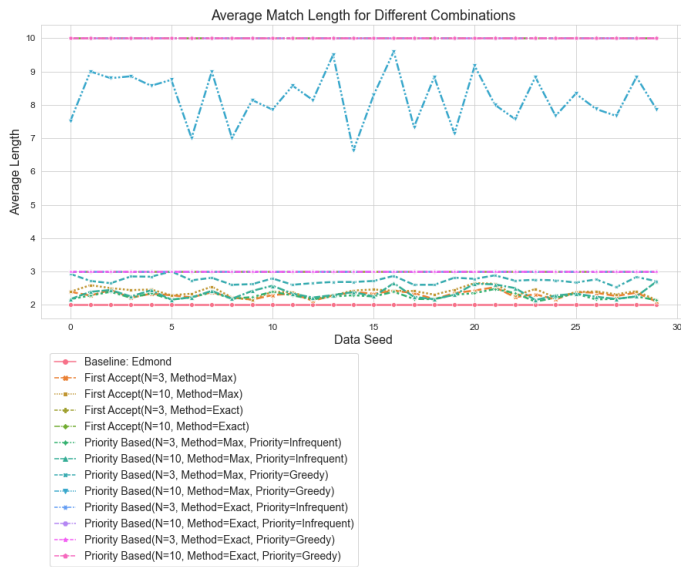
Meskipun algoritma pencarian pemetaan kecocokan N arah menghasilkan efisiensi pencocokan yang lebih tinggi, patut dicatat bahwa pertukaran-pertukaran N arah pada pemetaan kecocokan yang dihasilkan adalah pertukaran-pertukaran yang lebih sulit dieksekusi oleh rumah sakit karena prosedur transplantasi ginjal dari seluruh pasangan dalam suatu pertukaran harus dilakukan pada saat yang bersamaan. Implikasinya adalah pertukaran-pertukaran dengan jumlah arah yang sedikit lebih mudah untuk diterima dan dilakukan karena menggunakan lebih sedikit sumber daya rumah sakit. Oleh karena itu, eksperimen ini juga mencakup perbandingan mengenai algoritma mana yang mampu menghasilkan pemetaan kecocokan dengan panjang rata-rata dan maksimum siklus-siklus yang relatif rendah. Pada eksperimen ini, setiap kombinasi algoritma parameter diujikan dengan 30 data *seed* yang masing-masing mengandung 100 pasangan ketidakcocokan.

Seperti yang dapat dilihat pada Gambar 8, *Edmond's Algorithm* dan algoritma-algoritma dengan metode penggunaan nilai N eksak merupakan garis lurus pada panjang sama dengan N karena pertukaran-pertukaran yang diambil memiliki panjang yang sama dengan N . Hasil yang lebih menarik adalah meskipun parameter N adalah 10, seperti yang dapat dilihat



Gambar 8. Panjang Pertukaran Maksimum untuk Algoritma Berbeda

pada garis hijau dan kuning, panjang maksimum pertukaran secara signifikan lebih kecil dari 10.



Gambar 9. Panjang Pertukaran Rata-rata untuk Algoritma Berbeda

Seperti yang ada pada Gambar 8, pada Gambar 9, *Edmond's Algorithm* dan algoritma-algoritma dengan metode penggunaan nilai N eksak merupakan garis lurus pada panjang sama dengan N karena pertukaran-pertukaran yang diambil memiliki panjang yang sama dengan N . Secara umum, hampir semua algoritma yang menggunakan metode penggunaan nilai N maksimum menghasilkan siklus-siklus dengan panjang 2 hingga 3. Namun, algoritma *Priority-based Searching* dengan $N=10$, Method=Max, dan Priority=Greedy, menghasilkan rata-rata panjang pertukaran yang cukup tinggi.

IV. CONCLUSION

This paper shows that it is a viable option to use N -way match map searching algorithms instead of the usual two-way algorithms such as Edmond's Algorithm for Kidney Paired Donation, with N -way algorithms producing better matching efficiency and competitive execution time in comparison to the baseline, Edmond's Algorithm provided the same dataset. Generally speaking, the usage of N -way algorithms increases matching efficiency by 7.75%.

With the provided data, First Accept Searching algorithm have a more consistent performance in general compared to Priority-based Searching algorithm. However, with Priority-based Searching algorithm, the matching efficiency improvement is generally higher with the exception of when exact method to utilize N is being used instead of the maximum method. However, it is worth noting that priority-based searching algorithm may produce high matching efficiency with the compromise of high average matching length if greedy priority assignment method is being used. Therefore it can be concluded that First Accept Searching algorithm is the most ideal N -way match map searching algorithm to use in Kidney Exchange Program.

V. RELATED WORK

The idea of the solution, the existing algorithms, and the performance metrics being used as a performance measure are all obtained from the main reference paper, Web Based Decision Support System for Kidney Exchange by Samy Raja, Prasanna Devi S., and Suryaprakasa Rao K. in International Journal of Computer Applications [5]. In that paper, two-way match map searching algorithms are being used to tackle the problem of the humanely-impossible task of searching the most optimal match map out of an incompatible pairs pool, which becomes the baseline and the reference of the newer algorithms being implemented in this paper.

REFERENSI

- [1] Roth, A. E., Sonmez, T., Unver, M. U. (2005). Kidney Exchange. Quarterly Journal of Economics, 32.
- [2] Wiradarma, K. (2016, February 3). Transplantasi Ginjal di Indonesia: Pencapaian dan Hambatannya. Retrieved from klikdokter.com: <https://www.klikdokter.com/info-sehat/read/2697086/transplantasi-ginjal-di-indonesia-pencapaian-dan-hambatannya>
- [3] Roth, A. E., Sonmez, T., Unver, M. U., Delmonico, F. L., Saidman, S. L. (2006, September 18). Utilizing List Exchange and Nondirected Donation through 'Chain' Paired Kidney Donations. Retrieved from onlinelibrary.wiley.com: <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1600-6143.2006.01515.x>
- [4] Adrian, K. (2020, May 17). Berbagai Persiapan Donor Ginjal yang Perlu Anda Ketahui. Retrieved from alodokter.com: <https://www.alodokter.com/hal-hal-yang-harus-diperhatikan-sebelum-melakukan-donor-ginjal>
- [5] Raja, S., S., P. D., K., S. R. (2011). Web Based Decision Support System for Kidney. International Journal of Computer Applications (0975 – 8887), 9.
- [6] Chargé, S., Hodgkinson, K. (2017, January). Blood: the basics. Retrieved from profedu.blood.ca/: <https://profedu.blood.ca/en/transfusion/publications/blood-basics>.
- [7] Aprilano, W. D. (2021, January 20). Teknik Transplantasi Ginjal. Retrieved from alomedika.com: <https://www.alomedika.com/tindakan-medis/transplantasi/transplantasi-ginjal/teknik>

- [8] Nguyen, H. D., Williams, R. L., Wong, G., Lim, W. H. (2013, February 13). The Evolution of HLA-Matching in Kidney Transplantation. Retrieved from intechopen.com: <https://www.intechopen.com/books/current-issues-and-future-direction-in-kidney-transplantation/the-evolution-of-hla-matching-in-kidney-transplantation>
- [9] Tullis, T., Albert, B. (2013, June 3). Performance Metrics. Retrieved from sciencedirect.com: <https://www.sciencedirect.com/science/article/pii/B9780124157811000042>
- [10] Sedgewick, R., Wayne, K. (2020). Directed Graphs. In R. Sedgewick, K. Wayne, Algorithms, 4th Edition (p. 955). New Jersey: Princeton University.
- [11] Mehta, D. (2020, May 27). Detect Cycle in a Directed Graph. Retrieved from geeksforgeeks.org: <https://www.geeksforgeeks.org/detect-cycle-in-a-graph/>