

**ALGORITMA PENCARIAN UNTUK PEMETAAN KECOCOKAN
PADA PROGRAM PERTUKARAN GINJAL**

Laporan Tugas Akhir

Disusun sebagai syarat kelulusan tingkat sarjana

Oleh

LEONARDO

NIM : 13517048



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JUNI 2021**

**ALGORITMA PENCARIAN UNTUK PEMETAAN KECOCOKAN
PADA PROGRAM PERTUKARAN GINJAL**

Laporan Tugas Akhir

Oleh

LEONARDO

NIM : 13517048

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Telah disetujui dan disahkan sebagai Laporan Tugas Akhir
di Bandung, pada tanggal 29 Juni 2021

Pembimbing I,

Pembimbing II,

Dr. Fazat Nur Azizah, S.T., M.Sc.

NIP. 197902102009122001

Ardian Umam, S.T., M.Sc.

NIP. 119110074

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Pengerjaan dan penulisan Laporan Tugas Akhir ini dilakukan tanpa menggunakan bantuan yang tidak dibenarkan.
2. Segala bentuk kutipan dan acuan terhadap tulisan orang lain yang digunakan di dalam penyusunan laporan tugas akhir ini telah dituliskan dengan baik dan benar.
3. Laporan Tugas Akhir ini belum pernah diajukan pada program pendidikan di perguruan tinggi mana pun.

Jika terbukti melanggar hal-hal di atas, saya bersedia dikenakan sanksi sesuai dengan Peraturan Akademik dan Kemahasiswaan Institut Teknologi Bandung bagian Penegakan Norma Akademik dan Kemahasiswaan khususnya Pasal 2.1 dan Pasal 2.2.

Bandung, 29 Juni 2021



Leonardo

NIM 13517048

ABSTRAK

ALGORITMA PENCARIAN UNTUK PEMETAAN KECOCOKAN PADA PROGRAM PERTUKARAN GINJAL

Oleh

Leonardo

NIM : 13517048

Sebelum transplantasi ginjal dilakukan, resipien calon penerima ginjal dan juga calon pendonor harus bersifat cocok, yang berarti ginjal donor dapat bekerja pada tubuh resipien. Faktanya, sangat sering ketidakcocokan ini terjadi. Jika tidak cocok, pasangan donor-resipien yang pada awalnya seharusnya melakukan transplantasi akan dimasukkan ke program pertukaran ginjal, dimana pasangan donor-resipien akan dicocokkan dengan pasangan lain sehingga dapat dilakukan pendonoran silang. Untuk mempermudah, dibuatlah algoritma-algoritma untuk mencari pemetaan kecocokannya, yaitu algoritma pencarian pemetaan kecocokan. Algoritma yang ada terbatas pada pencocokan dua arah, menutup kemungkinan jika pencocokan tiga arah atau lebih dapat dilakukan.

Oleh karena itu, diimplementasikanlah algoritma pencarian pemetaan kecocokan bersifat N arah yang dapat menerima jumlah arah berapapun dengan batasan kekuatan operasional rumah sakit. Dengan memodifikasi algoritma pencarian pemetaan kecocokan yang sudah ada, dapat dibentuk algoritma-algoritma baru yang mampu menangani pencarian pemetaan kecocokan secara N arah.

Berdasarkan hasil pengujian menggunakan tiga ribu pasangan donor-resipien yang tidak cocok, didapatkan bahwa algoritma-algoritma N arah yang diimplementasikan berhasil memperoleh efisiensi pencocokan (*matching efficiency*) yang lebih tinggi dibandingkan saat menggunakan algoritma dua arah. Peningkatan efisiensi pencocokan yang didapatkan secara rata-rata mencapai 7.75% dibandingkan saat menggunakan algoritma yang sudah ada secara umum (*Edmond's Algorithm*).

Kata kunci: transplantasi ginjal, program pertukaran ginjal, algoritma pencarian pemetaan kecocokan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas kebesaran-Nya dan juga rahmat-Nya, penulis dapat mengerjakan, mengevaluasi, dan juga menyelesaikan Tugas Akhir penulis yang berjudul Implementasi Algoritma Pencarian untuk Pemetaan Pertukaran pada Program Pertukaran Ginjal dengan baik, dengan penuh semangat, tepat waktu, dan juga dengan dukungan dari orang-orang terdekat penulis. Adapun tujuan dari pengerjaan Tugas Akhir ini adalah untuk pemenuhan syarat kelulusan tingkat sarjana pada jurusan Teknik Informatika Institut Teknologi Bandung.

Penulis sangat bersyukur dengan adanya pihak-pihak yang telah membantu, memotivasi, dan juga mendukung penulis selama keberjalanan pembuatan Tugas Akhir yang dilaksanakan seluruhnya secara *online* ini. Penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Ibu Dr. Fazat Nur Azizah, S.T., M.Sc. dan juga Bapak Ardian Umam, S.T., M.Sc. selaku dosen-dosen pembimbing penulis yang telah memandu dan juga membimbing penulis dari awal pencarian ide, implementasi, hingga akhir validasi dan juga penulisan laporan.
2. Bapak Drs. Judhi Santoso, M.Sc. dan juga Ibu Fariska Zakhralativa Ruskanda, ST., M.T. selaku dosen-dosen penguji penulis yang telah memberikan kritik dan saran untuk tugas akhir yang dibuat oleh penulis.
3. Ibu Ginar Santika Niwanputri, ST., M.Sc., Bapak Adi Mulyanto, ST., MT., Bapak Nugraha Priya Utama, S.T., M.A., Ph.D., Ibu Latifa Dwiyantri, S.T., M.T., dan juga seluruh anggota tim tugas akhir yang telah mengadakan sarana perkembangan diri dan juga eksplorasi potensi dalam bentuk mata kuliah Tugas Akhir ini.
4. Seluruh dosen Institut Teknologi Bandung, terutama Teknik Informatika, yang telah mendidik dan juga melatih penulis dengan ilmu dan pengalaman dari awal kuliah hingga akhir waktu penulis berkuliah.

5. Keluarga penulis yang telah mendoakan penulis, yang telah mendanai perkuliahan penulis, yang telah mengajarkan penulis untuk menjadi pribadi berkarakter baik, yang telah menghidupkan penulis hingga sekarang, dan juga yang telah mendukung keberjalanan pengerjaan Tugas Akhir penulis.
6. Tim panitia ARKAVIDIA 7.0 yang dari Mei 2020 hingga Maret 2021 telah menghibur penulis selama ini dan juga yang telah berjalan bersama penulis dalam pembuatan ARKAVIDIA sebagai suatu acara, lomba, dan juga sebagai suatu sarana berkolaborasi nyata.
7. Seluruh asisten laboratorium Ilmu dan Rekayasa Komputasi yang telah membantu dan menemani penulis dalam pencarian ilmu, koneksi, dan juga uang jajan tambahan dalam kegiatan-kegiatan per-asisten-an.
8. Dua anggota lain grup Extra Virgin. Aditya Putra Santosa dan juga Vinsen Marselino Andreas yang telah menjadi teman penulis, mendengarkan cerita-cerita penulis, mengerjakan hal-hal aneh dan juga serius bersama penulis, dan juga menjadi orang-orang yang dapat diandalkan oleh penulis selama keberjalanan kuliah penulis.
9. Flute penulis yang selalu ada dan menemani penulis saat dibutuhkan dan yang senantiasa selalu siap untuk membuahkan suatu karya bersama penulis meskipun di tengah pandemi COVID-19.
10. Teman-teman penulis yang tidak dapat disebutkan namanya satu per satu yang ikut serta menemani penulis melalui proses pengerjaan ini.

Penulis berharap keberadaan dari Tugas Akhir ini dapat memberikan ilmu dan manfaat bagi siapapun yang membacanya. Semoga dengan adanya Tugas Akhir ini, dapat dijadikan semangat dan juga motivasi untuk berkarya.

Bandung, 29 Juni 2021

Penulis

DAFTAR ISI

ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR LAMPIRAN	viii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
DAFTAR SINGKATAN	xiii
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Tujuan	3
I.4 Batasan Masalah.....	4
I.5 Metodologi	4
I.6 Sistematika Pembahasan	6
BAB II STUDI LITERATUR	8
II.1 Transplantasi Ginjal	8
II.1.1 Golongan Darah	8
II.1.2 Uji Silang	9
II.1.3 Uji <i>Human Leucocyte Antigen</i> (HLA)	10
II.1.4 Serologi	10
II.2 Program-program Donor Ginjal	10
II.2.1 Program Transplantasi Donor Mayat	11
II.2.2 Program Transplantasi Donor Hidup	11
II.2.2.1 <i>Direct Donor</i> atau <i>Live Donation</i>	12

II.2.2.2	<i>Unrelated Live Donation</i>	12
II.2.2.3	<i>Kidney Paired Donation Program</i>	12
II.3	Graf Berarah.....	14
II.3.1	Algoritma Deteksi Siklus (<i>Cycle Detection</i>) pada Graf Berarah	14
II.4	Algoritma Pencarian Pemetaan Kecocokan	17
II.4.1	<i>Edmond's Algorithm</i>	17
II.4.2	Algoritma <i>First-accept Match Heuristic</i>	19
II.5	Metrik Kinerja.....	20
BAB III ALGORITMA PENCARIAN N ARAH UNTUK PEMETAAN PERTUKARAN PADA PROGRAM PERTUKARAN GINJAL		21
III.1	Masalah pada Algoritma Pencarian Pemetaan Kecocokan Dua Arah ..	21
III.2	Deskripsi Solusi	23
III.2.1	Data Medis Donor-Pasien	24
III.2.2	Pembangun Graf Kompatibilitas (<i>Compatibility Graph</i>).....	26
III.2.3	Algoritma Deteksi Siklus (<i>Cycle Detection</i>).....	27
III.2.4	Algoritma Pencarian Pemetaan Kecocokan	28
III.2.4.1	Algoritma <i>First-accept Searching</i>	30
III.2.4.2	Algoritma <i>Priority-based Searching</i>	31
BAB IV IMPLEMENTASI DAN EVALUASI		35
IV.1	Implementasi	35
IV.1.1	Pembangun Graf Kompatibilitas.....	35
IV.1.2	Algoritma Deteksi Siklus	36
IV.1.3	Algoritma Pencarian Pemetaan Kecocokan	36
IV.1.4	Aplikasi Pengujian	37

IV.1.4.1	Database.....	39
IV.1.4.2	Backend	40
IV.1.4.3	Frontend.....	42
IV.1.4.4	Batasan pada Aplikasi Pengujian.....	42
IV.2	Pengujian.....	44
IV.2.1	Pengujian menggunakan Aplikasi Pengujian.....	44
IV.2.2	Pengujian menggunakan Jupyter Notebook.....	44
IV.2.3	Lingkungan Pengujian (<i>Testing Environment</i>)	45
IV.3	Eksperimen Kombinasi Algoritma dan Parameter.....	45
IV.3.1	Perbandingan Waktu Eksekusi Setiap Algoritma	46
IV.3.2	Perbandingan Waktu Eksekusi untuk Jumlah Data yang Berbeda .	47
IV.3.3	Perbandingan Efisiensi Pencocokan berdasarkan Parameter Algoritma yang Digunakan	49
IV.3.3.1	Perbandingan Efisiensi Pencocokan berdasarkan Parameter Nilai N	49
IV.3.3.2	Perbandingan Efisiensi Pencocokan berdasarkan Parameter Metode Penggunaan Nilai N	55
IV.3.3.3	Perbandingan Efisiensi Pencocokan berdasarkan Parameter Metode Penentuan Prioritas	56
IV.3.4	Perbandingan Kombinasi Algoritma dan Parameter.....	58
BAB V	KESIMPULAN DAN SARAN	66
V.1	Kesimpulan	66
V.2	Saran.....	67
DAFTAR REFERENSI		68

DAFTAR LAMPIRAN

LAMPIRAN A. Kode Sumber Algoritma Pencarian Pemetaan Kecocokan ..	1
LAMPIRAN B. Hasil Eksperimen Perhitungan Matching Efficiency untuk Kombinasi Algoritma dan Parameter yang berbeda-beda	1
LAMPIRAN C. Hasil Eksperimen Rasio Perbandingan Algoritma N arah dengan Algoritma Baseline	1

DAFTAR GAMBAR

Gambar II.2.1 Two-way dan Three-way Exchange pada program KPD	13
Gambar II.3.1 Contoh Graf Berarah dengan 10 simpul dan 14 sisi	15
Gambar II.3.2 Siklus pada Graf Berarah.....	15
Gambar II.3.3 Flowchart Algoritma Pencarian Siklus untuk program KPD	16
Gambar II.4.1 Flowchart Edmond's Algorithm	18
Gambar II.4.2 Flowchart Algoritma First-accept Match Heuristic.....	19
Gambar III.1.1 Deteksi Sisi dibandingkan dengan Deteksi Siklus. Dapat terlihat pada kasus ini bahwa dengan menggunakan algoritma Deteksi Siklus, dapat tercakup tiga pasangan sehingga mampu menyelamatkan tiga pasien dibandingkan hanya dua pada Deteksi Sisi.....	22
Gambar III.1.2 Penggunaan Graf Tak Berarah dibandingkan dengan Penggunaan Graf Berarah.....	22
Gambar III.2.1 Block Diagram Rancangan Solusi	24
Gambar III.2.2 Contoh Graf Kompatibilitas untuk Data Contoh	26
Gambar III.2.3 Beberapa Siklus yang didapatkan dari Graf Kompatibilitas Contoh	27
Gambar III.2.4 Kasus dimana metode eksak unggul dibandingkan metode maksimum.....	29
Gambar III.2.5 Flowchart Algoritma First-Accept Searching	33
Gambar III.2.6 Flowchart Algoritma Priority-based Searching	34
Gambar IV.1.1 Class Diagram untuk Algoritma Pencarian Pemetaan Kecocokan	38
Gambar IV.1.2 Block Diagram Aplikasi Pengujian.....	39
Gambar IV.1.3 Tampilan Dashboard saat Membandingkan Lima Kombinasi ...	43

Gambar IV.3.1 Perbandingan Waktu Eksekusi Rata-rata Beberapa Kombinasi	47
Gambar IV.3.2 Perbandingan Waktu Eksekusi Rata-rata untuk Jumlah Data yang Berbeda	48
Gambar IV.3.3 Matching Efficiency untuk Nilai N yang berbeda dengan Algoritma First-Accept Maximum.....	50
Gambar IV.3.4 Matching Efficiency untuk Nilai N yang berbeda dengan Algoritma First-Accept Exact	51
Gambar IV.3.5 Matching Efficiency untuk Nilai N yang berbeda dengan Algoritma Priority-based Maximum Greedy	52
Gambar IV.3.6 Matching Efficiency untuk Nilai N yang berbeda dengan Algoritma Priority-based Maximum Infrequent	53
Gambar IV.3.7 Matching Efficiency untuk Nilai N yang berbeda dengan Algoritma Priority-based Exact Greedy	54
Gambar IV.3.8 Matching Efficiency untuk Nilai N yang berbeda dengan Algoritma Priority-based Exact Infrequent.....	55
Gambar IV.3.9 Perbandingan Metode Penggunaan Nilai N pada Algoritma First Accept Search.....	57
Gambar IV.3.10 Perbandingan Metode Penggunaan Nilai N pada Algoritma Priority Based Search Greedy	58
Gambar IV.3.11 Perbandingan Metode Penggunaan Nilai N pada Algoritma Priority Based Search Infrequent	59
Gambar IV.3.12 Perbandingan Metode Penentuan Prioritas pada Algoritma Priority Based Search Maximum	60
Gambar IV.3.13 Perbandingan Metode Penentuan Prioritas pada Algoritma Priority Based Search Exact.....	61
Gambar IV.3.14 Rasio Perbandingan Algoritma-algoritma N arah dengan Baseline: Edmond	63

Gambar IV.3.15 Arah Pencocokan Maksimum untuk Kombinasi yang Berbeda⁶⁴

Gambar IV.3.16 Arah Pencocokan Rata-rata untuk Kombinasi yang Berbeda .. 65

DAFTAR TABEL

Tabel II.1.1 Kompatibilitas Golongan Darah Penerima dan Pemberi Donor Ginjal (Raja, dkk., 2011).....	9
Tabel III.2.1 Struktur Data Medis Donor dan Pasien yang Dibutuhkan.....	25
Tabel III.2.2 Contoh Data Medis Donor-Pasien	25
Tabel IV.1.1 Contoh Graf Kompatibilitas dengan Representasi Adjacency List	36
Tabel IV.1.2 Contoh List Siklus dengan Representasi List of Lists	36
Tabel IV.1.3 Contoh Keluaran Algoritma Pencarian Pemetaan Kecocokan	37
Tabel IV.1.4 Distribusi Data Medis Donor dan Pasien.....	41
Tabel IV.1.5 Struktur Data Hasil Terbaik pada Basis Data	42
Tabel IV.3.1 Rata-rata Rasio Peningkatan Matching Efficiency Algoritma N Arah dibandingkan dengan Algoritma Baseline	62

DAFTAR SINGKATAN

KPD	: <i>Kidney Paired Donation</i>
KEP	: <i>Kidney Exchange Program</i>
HLA	: <i>Human Leucocyte Antigen</i>
MHC	: <i>Major Histocompatibility Complex</i>
PRA	: <i>Panel-Reactive Antibody</i>
DFS	: <i>Depth-First Search</i>
HTML	: <i>Hyper-Text Markup Language</i>
CSS	: <i>Cascading Style Sheets</i>
CRUD	: <i>Create, Read, Update, Delete</i>
OS	: <i>Operating System</i>
RAM	: <i>Random Access Memory</i>
CPU	: <i>Central Processing Unit</i>

BAB I

PENDAHULUAN

Bab pendahuluan berisikan mengenai alasan secara umum mengapa tugas akhir ini dilakukan. Bab ini terdiri dari enam subbab, yaitu latar belakang, rumusan masalah tujuan, batasan masalah, metodologi, dan diakhiri dengan sistematika pembahasan untuk setiap bab-bab lainnya pada buku tugas akhir ini.

I.1 Latar Belakang

Di dunia kesehatan, terdapat banyak hal yang dapat menimbulkan masalah. Sebagian besar masalah yang terjadi dapat menelan korban jiwa dari pasien-pasien yang membutuhkan pertolongan. Salah satu dari masalah yang dapat timbul dari bidang kesehatan ada pada cabang ilmu kedokteran nefrologi, atau lebih akrab disebut dengan ilmu kedokteran ginjal.

Ginjal manusia merupakan salah satu organ terpenting pada sistem ekskresi manusia. Ginjal manusia berfungsi untuk mengeluarkan zat-zat sisa metabolisme dan juga limbah kotoran di tubuh melalui urin. Akan tetapi, tidak semua orang cukup beruntung untuk memiliki ginjal yang sehat, beberapa orang bahkan tidak memiliki satupun ginjal yang fungsional di tubuhnya. Tidak memiliki ginjal yang berfungsi tentunya merupakan suatu masalah besar pada manusia. Dari suatu masalah, muncul suatu solusi. Solusi itu adalah transplantasi ginjal.

Setiap manusia dilahirkan dengan dua buah ginjal, dan pada umumnya, manusia dapat bertahan hidup dengan hanya satu buah ginjal sebagai pusat dari sistem ekskresinya. Oleh karena itu, salah satu dari ginjal sehat seorang manusia dapat didonasikan dan digunakan oleh seorang pasien yang memiliki kebutuhan akan ginjal yang sehat dan fungsional. Program ini disebut juga sebagai program transplantasi ginjal.

Terlepas dari adanya program transplantasi ginjal, sangat banyak pasien yang membutuhkan transplantasi ginjal tidak mendapatkan donasi yang mereka butuhkan. Di Indonesia sendiri, sekitar 100,000 orang pasien memerlukan ginjal

transplantasi setiap tahunnya, namun hanya sekitar 20% dari total pasien tersebut yang dapat tertolong dari program transplantasi ini. Hal ini disebabkan oleh masalah-masalah yang timbul seperti masalah finansial, masalah regulasi, dan salah satu yang paling sering terjadi, masalah ketersediaan (Wiradarma, 2016).

Masalah ketersediaan umumnya terjadi karena tidak adanya orang lain yang bersedia untuk mendonorkan ginjalnya. Lalu jika seseorang sudah bersedia untuk mendonorkan ginjalnya, orang yang bersangkutan mungkin tidak memiliki ginjal yang cukup sehat untuk didonasikan. Lalu jika orang yang sudah bersedia memiliki ginjal yang cukup sehat untuk didonasikan, ginjal tersebut belum tentu cocok untuk ditanamkan pada tubuh pasien yang dituju. Kriteria cocok atau tidaknya suatu ginjal pendonor dengan pasien calon penerima transplantasi ginjal adalah cocok secara golongan darah dan sistem imun pasien tidak menolak ginjal dari pendonor. Kedua kriteria ini akan dijelaskan lebih lanjut pada tahap studi literatur.

Banyak muncul keresahan baik dari sisi pendonor maupun dari sisi pasien mengenai ketidakcocokan ini. Jika ginjal tidak cocok untuk ditanamkan, maka donor tidak dapat dilakukan. Oleh karena itu, muncullah *Kidney Exchange Program* (KEP) atau disebut juga program pertukaran ginjal. Pada program ini, pasangan pendonor-pasien yang tidak cocok dapat dicocokkan dengan sumber-sumber ginjal transplantasi lainnya untuk mendapatkan kecocokan sehingga donor ginjal tetap dapat dilakukan alhasil lebih banyak pasien yang dapat diselamatkan karena berhasil menerima transplantasi ginjal.

Untuk melakukan program pertukaran ginjal ini tidaklah mudah, diperlukan banyak sumber daya untuk dapat mencari pemetaan kecocokan yang optimal untuk para pasien. Dikarenakan sangat banyaknya pasangan donor-pasien yang ada, pencarian pemetaan kecocokan ini sangat sulit untuk dilakukan oleh manusia.

I.2 Rumusan Masalah

Di era digital ini, sudah banyak sistem dan algoritma yang menyelesaikan masalah pencarian yang telah dijelaskan sebelumnya. Implementasi algoritma pencarian kecocokan untuk program pertukaran ginjal sudah sangat banyak dilakukan dan

diaplikasikan oleh instansi-instansi kedokteran di dunia. Algoritma pencarian pemetaan kecocokan harus memenuhi dua buah kriteria, yang pertama, dari segi jumlah kecocokan. Algoritma pencarian pemetaan kecocokan harus dapat memperoleh pemetaan kecocokan yang menghasilkan jumlah kecocokan yang maksimal dari sejumlah data yang terbatas karena semakin besar jumlah kecocokannya, semakin banyak pasien yang dapat terselamatkan. Algoritma pencarian pemetaan kecocokan juga harus dapat memperoleh hasil tersebut secara cepat. Semakin cepat hasil diperoleh dari algoritma, semakin cepat transplantasi dapat dilakukan, sehingga pasien dapat memperoleh ginjal transplan sebelum kondisi semakin memburuk. Dari algoritma pencarian pemetaan kecocokan yang sudah ada, *Edmond's algorithm*, dan juga algoritma pencarian pemetaan kecocokan bersifat N arah yang ada, terdapat dua buah masalah yaitu:

1. Pada aplikasi-aplikasi yang pernah dibuat, algoritma yang digunakan adalah *Edmond's Algorithm* (Raja dkk., 2011), yang tidak mampu mencari solusi jika dilakukan pertukaran yang bersifat lebih dari dua arah (*two-way exchange*). Kemampuan dan fleksibilitas sistem untuk mencari kecocokan-kecocokan saat pertukaran dilakukan secara N arah (*n-way exchange*) harapannya dapat meningkatkan efisiensi pencocokan dari pasangan-pasangan donor-resipien secara drastis.
2. Algoritma N arah yang sudah ada pada proposal-proposal ilmiah seperti pada *paper* rujukan utama, Raja dkk., 2011, belum dicobakan langsung pada data nyata. Sehingga algoritma yang ada tidak dijamin optimal dan efisien. Seperti yang telah dijelaskan, efisiensi yang dimaksudkan disini adalah banyaknya jumlah kecocokan pada pemetaan kecocokan dan juga waktu eksekusi dari algoritma dari mulai berjalan hingga selesai.

I.3 Tujuan

Berdasarkan rumusan masalah yang disampaikan, tujuan dari pengerjaan Tugas Akhir ini adalah untuk membuat algoritma pencarian pemetaan kecocokan untuk *Kidney Exchange Program* secara N arah yang didasarkan pada algoritma *Edmond's* dan algoritma *First-accept Match Heuristic* (Raja dkk., 2011). Algoritma pencarian

pemetaan kecocokan N arah diharapkan dapat meningkatkan efisiensi pencocokan yang dapat terjadi.

Target lain dari pengerjaan Tugas Akhir ini adalah untuk dapat mencari algoritma pencarian pemetaan kecocokan yang efisien untuk penyelesaian masalah. Seperti yang telah dijelaskan pada subbab I.2 tahap rumusan masalah, efisien dinilai berdasarkan dua hal, yaitu banyaknya kecocokan yang didapat dari total data yang ada, dan juga waktu eksekusi algoritma yang singkat.

I.4 Batasan Masalah

Batasan-batasan dari permasalahan ini adalah:

1. Algoritma-algoritma pencarian kecocokan diimplementasikan untuk pasangan donor hidup dan resipien yang tidak cocoknya kepada pasangan donor hidup dan resipien yang tidak cocok lainnya saja. Pada pelaksanaan tugas akhir ini, tidak dilakukan implementasi untuk mencari kecocokan antara pasangan ketidakcocokan dengan donor ginjal *cadaver* (ginjal yang berasal dari mayat).
2. Selain itu, algoritma-algoritma yang dibuat hanya akan bekerja pada data statis. Sehingga penambahan data tidak dapat dilakukan di tengah jalannya algoritma.

I.5 Metodologi

Pengerjaan tugas akhir ini akan dipecah-pecah menjadi beberapa tahap besar. Pada setiap tahap besar terdapat suatu keluaran yang bermakna yang dapat dibawa ke tahapan besar berikutnya. Berikut adalah tahapan-tahapan secara berurutan yang akan dilalui dalam proses pengerjaan tugas akhir:

1. Analisis Masalah dan Rancangan Solusi

Pada tahap ini, dilakukan analisis secara mendalam mengenai akar permasalahan yang akan dihadapi dan juga analisis mengenai solusi-solusi yang mungkin diimplementasikan. Dilakukan juga rancangan secara lebih

menyeluruh mengenai algoritma-algoritma pencarian pemetaan kecocokan yang akan diimplementasikan.

2. Implementasi Algoritma-algoritma Pencarian Pemetaan Kecocokan

Pada tahap ini, diimplementasikanlah beberapa buah algoritma pencarian pemetaan kecocokan yang dapat memproses data pasangan donor-resipien transplantasi ginjal yang tidak cocok dan mencocokkannya dengan pasangan-pasangan lain agar didapatkan kecocokan paling banyak. Algoritma-algoritma tersebut dapat memproses data tersebut dan mengembalikan kumpulan kecocokan antarpasangan pendonor-pasien.

3. Pembuatan Lingkungan Pengujian dan Evaluasi

Untuk dilakukan pengujian, pada tahap ini, dikumpulkan data mengenai pasangan pendonor dan pasien. Data dapat diperoleh dari berbagai sumber, seperti rumah sakit lokal, data *open source*, maupun penulis *paper* lain. Data yang didapat berisikan tentang informasi-informasi medis dari pasien dan juga pendonor. Baik itu kondisi kesehatan, golongan darah, maupun juga *Panel-Reactive Antibody* (PRA) dari pasien calon penerima donor ginjal yang bersangkutan. Pada tahap ini, diusahakan untuk dapat memperoleh data yang bersifat baru dan berasal dari Indonesia, agar hasil yang didapatkan sistem akurat dan relevan untuk masyarakat.

Data yang telah diperoleh pada proses sebelumnya diolah agar bersih, sesuai format, rapi, dan juga konsisten. Data dibentuk menjadi bentuk yang dapat dimasukkan ke algoritma-algoritma pencarian untuk mendapatkan hasil. Terdapat beberapa teknik pengolahan data seperti pembersihan, reduksi, dan juga transformasi.

Menggunakan algoritma-algoritma pencarian pemetaan kecocokan yang telah diimplementasikan pada tahap sebelumnya, dibuatlah suatu lingkungan pengujian seputar algoritma-algoritma pencarian tersebut dengan kebutuhan-kebutuhan lainnya agar algoritma pencarian pemetaan kecocokan dapat berjalan dengan baik. Lingkungan pengujian ini digunakan

agar tidak terjadi *bias* secara teknologi terhadap hasil evaluasi dan pengujian saat eksperimen dilakukan.

Pada tahap ini, hasil dari pencarian pemetaan kecocokan beberapa kombinasi algoritma dan parameter-parameternya akan dianalisis dan dievaluasi. Terdapat dua metrik yang dievaluasi, yaitu banyaknya kecocokan yang terjadi, dan juga waktu eksekusi algoritma. Semakin banyak kecocokan yang terjadi maka semakin banyak pasien yang dapat menerima ginjal donor. Semakin cepat algoritma berjalan berarti semakin cepat pasien mendapatkan ginjal donor.

I.6 Sistematika Pembahasan

Pada bab 1 buku ini, dijelaskan mengenai latar belakang, masalah-masalah yang ada yang ingin diselesaikan, implikasinya untuk tujuan, batasan-batasan dari *scope* permasalahan yang akan diselesaikan, dan juga linimasa kasar metode-metode penyelesaian masalahnya.

Pada bab 2 buku ini, dijelaskan dasar-dasar teori yang bersifat membangun pengertian mengenai topik permasalahan dan juga solusi penyelesaian yang direncanakan dan juga diimplementasikan. Penjelasan dilakukan pada bidang-bidang yang di luar pengetahuan umum. Penjelasan mengenai transplantasi ginjal, program-program donor ginjal, hingga teori-teori informatika seperti graf berarah hingga algoritma pencarian pemetaan kecocokan.

Pada bab 3 buku ini, dideskripsikan secara rinci mengenai analisis masalah yang kritis dan juga analisis dari alternatif-alternatif solusi, pemilihan solusi yang dilakukan. Pada bab ini, dicantumkan juga penjelasan untuk rancangan solusi umum yang akan dibuat.

Pada bab 4, diuraikan implementasi untuk lingkungan pengujian dan juga evaluasi-evaluasi yang dilakukan. Evaluasi-evaluasi ini dilakukan seadil mungkin untuk setiap teknologi yang digunakan agar hasil yang dikeluarkan konsisten dan tidak terdapat bias yang disebabkan oleh *hardware*. Pada bab ini juga dijelaskan

mengenai aplikasi sederhana yang digunakan untuk mempermudah pengujian yang dilakukan.

Pada bab 5, dicantumkan kesimpulan-kesimpulan yang didapat dari keseluruhan proses pelaksanaan pengerjaan tugas akhir. Selain kesimpulan, dicantumkan juga saran-saran yang penulis miliki bagi pihak-pihak yang ingin melakukan riset lebih dalam mengenai topik ini.

BAB II

STUDI LITERATUR

Pada bab ini, dijabarkan mengenai landasan teori yang digunakan pada pengerjaan tugas akhir ini dimulai dari penjelasan mengenai transplantasi ginjal hingga algoritma yang sudah ada sebelumnya untuk penyelesaian masalah. Bab ini terdiri dari lima buah subbab besar, yaitu subbab yang menjelaskan mengenai transplantasi ginjal dan juga *pre-requisite* agar transplantasi ginjal dapat dilakukan, subbab mengenai program-program donor ginjal yang sudah ada, subbab mengenai graf berarah, subbab mengenai algoritma pencarian pemetaan kecocokan yang berisikan mengenai algoritma-algoritma yang sudah ada untuk program pertukaran ginjal, dan diakhiri dengan subbab mengenai metrik kinerja.

II.1 Transplantasi Ginjal

Transplantasi ginjal adalah perawatan yang paling direkomendasikan untuk penyakit-penyakit ginjal yang serius (Roth dkk., 2005). Sebelum melakukan transplantasi ginjal, perlu dilakukan dua macam pengetesan darah baik dari pihak pemberi donor maupun pihak penerima donor. Hal-hal yang harus dilakukan ialah pengetesan kecocokan golongan darah kedua pihak, uji silang untuk mengetahui ada atau tidaknya antibody yang berpotensi melawan sel-sel ginjal pendonor, tes *Human Leucocyte Antigen* (HLA) untuk mengetahui *Panel-Reactive Antibody* (PRA) dari pasien penerima donor, dan juga uji serologi untuk mengetahui adanya infeksi virus HIV pada pendonor.

II.1.1 Golongan Darah

Pengetesan golongan darah adalah tes darah yang pertama dilakukan oleh penerima donor dan juga pemberi donor. Setelah sampel darah dari kedua pihak didapatkan, kecocokan dari golongan darah kedua pihak akan dibandingkan tanpa mempertimbangkan rhesus karena dianggap tidak relevan pada pencocokan ginjal (Raja dkk., 2011). Kecocokan atau kompatibilitas dari darah penerima dan juga pemberi donor dapat dilihat pada tabel II.1.1.

Tabel II.1.1 Kompatibilitas Golongan Darah Penerima dan Pemberi Donor Ginjal (Raja, dkk., 2011)

Penerima \ Pemberi	O	A	B	AB
O	1	0	0	0
A	1	1	0	0
B	1	0	1	0
AB	1	1	1	1

Pada Tabel II.1.1, angka 1 menyatakan bahwa pasangan pemberi dan penerima donor kompatibel. Sebaliknya, angka 0 menyatakan bahwa pasangan pemberi dan penerima donor tidak kompatibel. Pada tabel II.1.1, terdapat suatu keunikan yang terjadi untuk pemberi donor bergolongan darah O, yaitu ginjal donor yang bersifat cocok ke penerima donor bergolongan darah apapun. Pemberi donor bergolongan darah O disebut juga sebagai *Universal Donor* (Chargé & Hodgkinson, 2017). Ada pula penerima donor yang bersifat universal. Berdasarkan tabel II.1.1, dapat dilihat bahwa penerima donor dengan golongan darah AB kompatibel dengan pemberi donor dari golongan darah apapun. Maka penerima donor dengan golongan darah AB disebut juga *Universal Recipient* (Chargé & Hodgkinson, 2017).

II.1.2 Uji Silang

Jika golongan darah dari kedua pihak cocok, maka kedua pihak dapat melaksanakan tes darah yang kedua, yaitu uji silang (Adrian, 2020). Pengetesan ini dilakukan untuk mencegah serangan oleh sistem imun penerima donor kepada ginjal transplan yang diberikan pendonor. Penyerangan dilakukan saat ginjal pendonor dianggap sebagai benda asing yang harus dibunuh oleh sistem imun penerima donor (Aprilano, 2021).

Terdapat dua buah hasil yang mungkin dari tes ini, yaitu positif dan negatif. Apabila hasil tes positif, maka resipien memiliki antibodi terhadap donor. Sebaliknya,

apabila hasil tes negatif, maka resipien tidak memiliki antibodi terhadap donor (Aprilano, 2021).

II.1.3 Uji *Human Leucocyte Antigen* (HLA)

Uji HLA dilakukan untuk mengetahui tipe jaringan dari pendonor dan resipien (Aprilano, 2021). HLA sendiri adalah *Major Histocompatibility Complex* (MHC) milik manusia. HLA berperan untuk mengontrol respon sistem imun manusia melalui antigen dan untuk membedakan sel-sel asing dan sel-sel tubuh (Nguyen dkk., 2013).

Ginjal yang boleh menjadi ginjal donor untuk suatu resipien hanyalah ginjal dari pendonor yang dapat menghasilkan persentase *Panel Reactive Antibody* (PRA) terkecil saat direaksikan dengan sel-sel darah resipien. PRA itu sendiri adalah persentase dari seberapa sering suatu reaksi penolakan terjadi dari sekian buah tes. Persentase ini berfungsi untuk menyatakan seberapa besar kemungkinan terjadi rejeksi yang akan terjadi saat ginjal donor dicangkokkan ke pasien (Nguyen dkk., 2013). Pada zaman modern ini, pengetesan HLA bersifat opsional karena sudah majunya obat-obatan imunosupresan yang berfungsi untuk meredam reaksi penolakan dari sel-sel darah pasien (Raja dkk., 2011).

II.1.4 Serologi

Pemeriksaan ini dilakukan untuk melihat ada atau tidaknya infeksi virus-virus berbahaya seperti HIV, hepatitis, dan *cytomegalovirus*. Pemeriksaan ini dilakukan agar terapi pencegahan yang tepat dapat diberikan setelah transplantasi dilakukan. Pemeriksaan ini dilakukan ke semua calon pendonor agar tidak menularkan penyakit ke resipien penerima donor (Aprilano, 2021).

II.2 Program-program Donor Ginjal

Tidak semua pasien yang membutuhkan transplantasi ginjal memiliki pendonor yang dapat memberikan ginjal. Untuk menanggulangi hal ini, dibuatlah beberapa program transplantasi ginjal (Raja dkk., 2011).

II.2.1 Program Transplantasi Donor Mayat

Program ini adalah program donor dan transplantasi ginjal untuk ginjal yang berasal dari orang-orang yang sudah meninggal. Untuk setiap ginjal dari mayat tersedia dan dapat digunakan untuk transplantasi, maka dilakukan pencocokan dengan daftar pasien berkebutuhan donor ginjal yang telah mendaftarkan diri untuk menerima ginjal mayat. Perencanaan untuk transplantasi dilakukan saat didapatkan pasien pertama yang memiliki kecocokan sempurna dengan ginjal donor dari mayat. Prioritas dari setiap pasien untuk mendapatkan donor diatur menggunakan dua hal, yaitu waktu tunggu dan juga kecocokan golongan darah dengan mayat pendonor. Meskipun terdengar efisien dan sangat bagus, program ini memiliki beberapa kelemahan, yaitu:

1. Hanya sedikit mayat yang secara medis dapat menjadi pendonor. Sehingga jumlah ginjal yang didapatkan tetap lebih kecil dibandingkan dengan pasien yang membutuhkan donor.
2. Banyak ginjal donor berkualitas rendah karena diambil dari orang mati. Sehingga tidak sedikit pasien yang menolak ginjal yang dipilih meskipun memiliki kecocokan yang sempurna.
3. Tingkat kelangsungan hidup transplan lebih rendah dibandingkan dengan transplantasi donor hidup.

Kelemahan-kelemahan dari program transplantasi donor mayat diatasi oleh program transplantasi ginjal donor hidup (Raja dkk., 2011).

II.2.2 Program Transplantasi Donor Hidup

Donor hidup merupakan sumber transplantasi ginjal yang terus menerus meningkat (Roth, dkk., 2006). Tidak seperti program transplantasi donor mayat, ginjal yang didapatkan dan digunakan dalam program transplantasi ini adalah ginjal dari orang hidup. Dibandingkan program transplantasi ginjal donor mayat, tingkat kelangsungan hidup dari penerima donor relatif lebih tinggi dan secara ketersediaan lebih banyak sehingga waktu tunggu pasien lebih rendah (Raja dkk., 2011).

Meskipun lebih baik dari program transplantasi ginjal donor mayat, terkadang beberapa masalah muncul seperti ketidakcocokan antara pendonor dan penerima.

Banyak orang dengan keinginan untuk mendonor tidak dapat mendonor dikarenakan adanya ketidakcocokan kepada calon penerima donor yang dituju. Oleh karena itu, telah dibuat beberapa program untuk meningkatkan kemungkinan pasien berkebutuhan ginjal transplan untuk mendapatkan transplantasi ginjal (Raja dkk., 2011).

II.2.2.1 *Direct Donor atau Live Donation*

Program ini ditujukan untuk donor ginjal yang berasal dari anggota keluarga dan kerabat dari pasien berkebutuhan. Seperti yang telah dituangkan sebelumnya, kemungkinan untuk kecocokan lebih tinggi apabila pendonor berasal dari keluarga yang sama dengan penerima donor. Donor ini dilakukan salah satunya untuk pasien-pasien yang mengantre untuk donor ginjal mayat. Donor dilakukan apabila kecocokan antara pemberi donor dan penerima donor ginjal bersifat sempurna. Donor dilakukan secara langsung (*live*) (Raja dkk., 2011).

II.2.2.2 *Unrelated Live Donation*

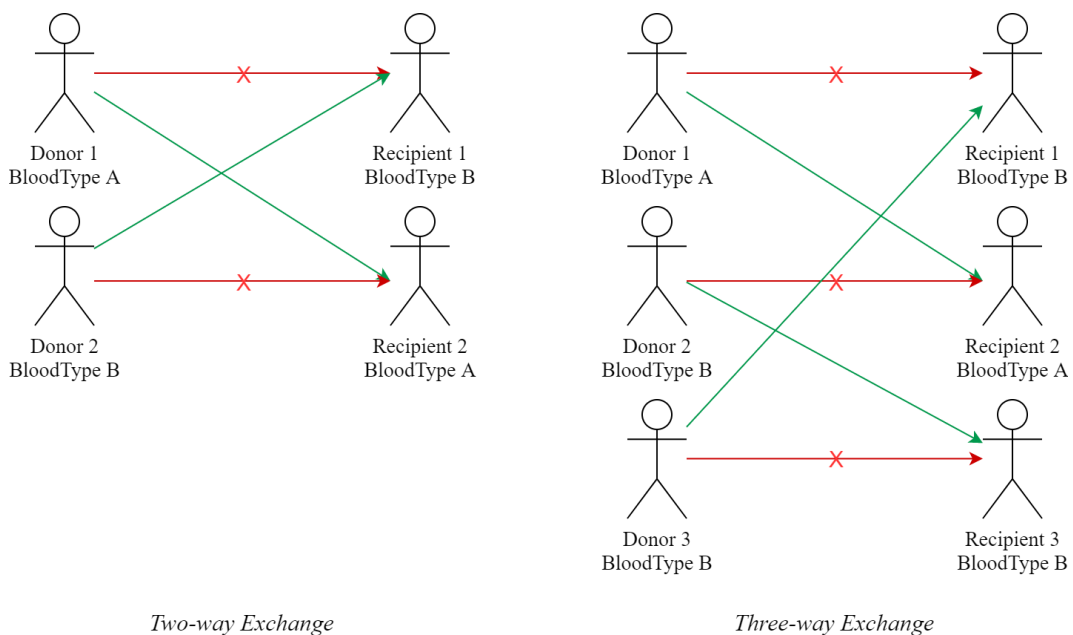
Sama seperti program *Direct Donor*, program ini dilakukan secara langsung. Namun untuk pendonor yang tidak berasal dari keluarga yang sama dari pasien calon penerima donor. Seperti program sebelumnya, donor juga dilakukan saat terdapat kecocokan sempurna antara pemberi donor dan penerima donor ginjal (Raja dkk., 2011).

II.2.2.3 *Kidney Paired Donation Program*

Seperti yang telah dijelaskan sebelumnya, pada nyatanya, tidak jarang terjadi ketidakcocokan antara pendonor dan juga pasien calon penerima bahkan meskipun kedua pihak berasal dari keluarga yang sama. Oleh karena itu, donor secara langsung tidak dapat dilakukan. Pasien akan membutuhkan ginjal dari donor lain yang terkadang sulit ditemukan. Untuk menyelesaikan masalah ini, pendonor dan pasien calon penerima dapat melakukan registrasi pada program *Kidney Paired Donation (KPD) Exchange* atau biasa disingkat *Paired Exchange*. Program ini berfungsi untuk melakukan *pairing* atau pemasangan pendonor dan penerima donor yang cocok. Program ini memungkinkan penerima untuk mendapatkan ginjal dari

donor lain yang ginjalnya tidak cocok untuk penerima yang seharusnya dan juga sebaliknya (Raja dkk., 2011).

Pada program KPD terdapat beberapa cara melakukan pertukaran seperti *Two-way exchange* dan *Three-way exchange*. Tidak harus dua atau tiga arah, pertukaran dapat dilakukan hingga N arah. Pada gambar II.2.1 bagian kiri, pendonor 1 tidak dapat memberikan ginjalnya kepada penerima 1 dikarenakan golongan darah yang tidak kompatibel. Pada saat yang sama, pendonor 2 tidak dapat memberikan ginjalnya pula kepada penerima 2 karena perbedaan golongan darah. Saat dua pasangan yang masing-masing memiliki ginjal donor yang tidak cocok melakukan pertukaran sehingga terjadi kecocokan disebut sebagai *Two-way exchange*. Jika kedua pasangan pendonor-penerima tetap tidak dapat melakukan pertukaran secara *two-way*, maka kedua pasangan ini akan dicocokkan dengan pasangan donor-resipien lain. Pertukaran ini disebut juga dengan *Three-way exchange*, yang ilustrasinya dapat dilihat pada Gambar II.2.1 bagian kanan.



Gambar II.2.1 *Two-way* dan *Three-way Exchange* pada program KPD

Pada contoh di Gambar II.2.1, Donor 3 tidak dapat memberikan ginjalnya kepada pasien 3 karena nilai PRA yang tinggi. Namun pasien 3 mendapatkan kecocokan dengan pendonor 2 sehingga pendonor 2 memberikan ginjalnya kepada pasien 3 dan ginjal dari pendonor 3 dapat diberikan kepada pasien 1.

Pencocokan untuk pertukaran ini dapat terjadi hingga N arah. Kekurangan dari pertukaran N arah ini adalah sumber daya kedokteran, kebutuhan infrastruktur, dan fasilitas logistik yang besar karena pertukaran ini harus dilakukan secara bersamaan. Program KPD ini telah secara efektif berhasil meningkatkan jumlah pasien yang mendapatkan transplantasi ginjal di dunia (Raja dkk., 2011).

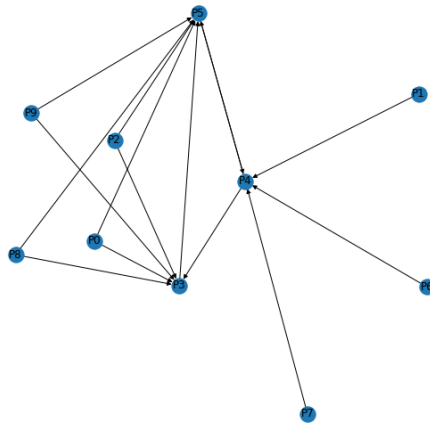
Terdapat suatu modifikasi dari program KPD yang dinamakan *List Exchange Program*. Pada program *List Exchange* ini, selain dicocokkan dengan pasangan-pasangan ketidakcocokan lainnya, pasangan-pasangan donor-pasien yang tidak cocok juga dipasangkan dengan ginjal-ginjal *cadaver* (Raja dkk., 2011).

II.3 Graf Berarah

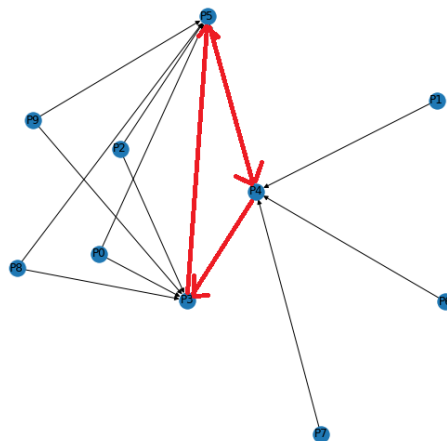
Seperti yang dapat dilihat pada Gambar II.2.1, pertukaran pada program KPD pada umumnya dapat direpresentasikan dalam bentuk graf. Dalam teori graf, graf berarah (*directed graph*) merupakan suatu graf yang terdiri dari sekumpulan simpul (*vertices*) dan dihubungkan oleh sisi (*edges*), dimana setiap sisi memiliki arah yang menunjukkan arah pergerakan yang dapat dilakukan dari suatu simpul ke simpul lainnya (Sedgewick & Wayne, 2020).

II.3.1 Algoritma Deteksi Siklus (*Cycle Detection*) pada Graf Berarah

Algoritma pencarian siklus merupakan algoritma yang digunakan untuk mencari keberadaan suatu jalur pergerakan pada graf yang dimulai dan selesai pada simpul yang sama (Mehta, 2020). Pada program pertukaran ginjal, siklus yang terjadi merupakan arah dari pertukaran ginjal yang dilakukan. Jika siklus yang terjadi adalah $A \rightarrow B \rightarrow C \rightarrow A$, maka pendonor A akan memberikan ginjalnya kepada pasien B, pendonor B akan memberikan ginjalnya kepada pasien C, dan pendonor C akan memberikan ginjalnya kepada pasien A. Pertukaran yang terjadi bersifat 3 arah (*three-way exchange*).

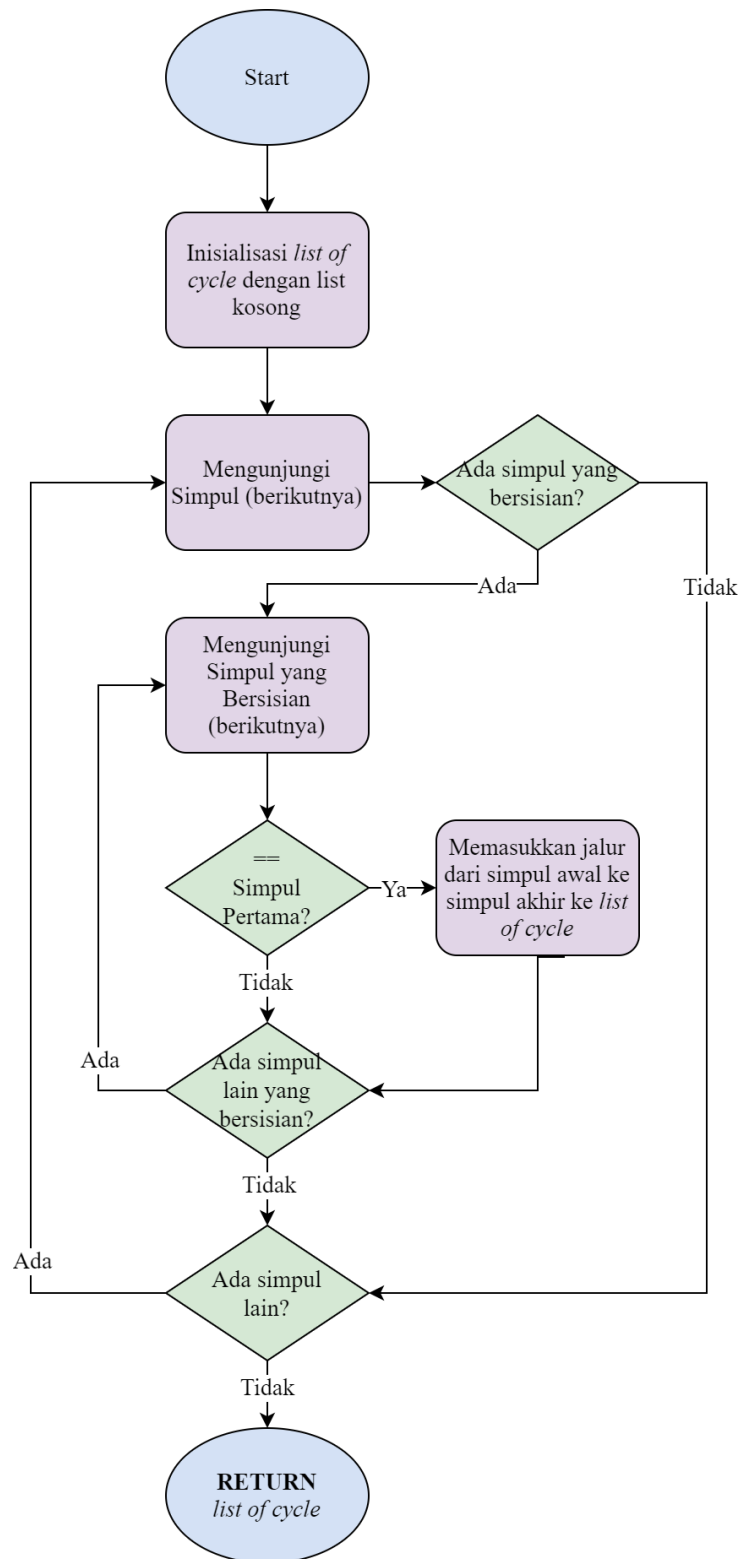


Gambar II.3.1 Contoh Graf Berarah dengan 10 simpul dan 14 sisi



Gambar II.3.2 Siklus pada Graf Berarah

Pada program pertukaran ginjal, algoritma pencarian siklus dibuat untuk menemukan semua siklus yang terdapat pada graf keterhubungan kecocokan antarpasangan ketidakcocokan yang telah terbuat dari data. *Flowchart* dari algoritma pencarian siklus dapat dilihat pada Gambar II.3.3.



Gambar II.3.3 Flowchart Algoritma Pencarian Siklus untuk program KPD

II.4 Algoritma Pencarian Pemetaan Kecocokan

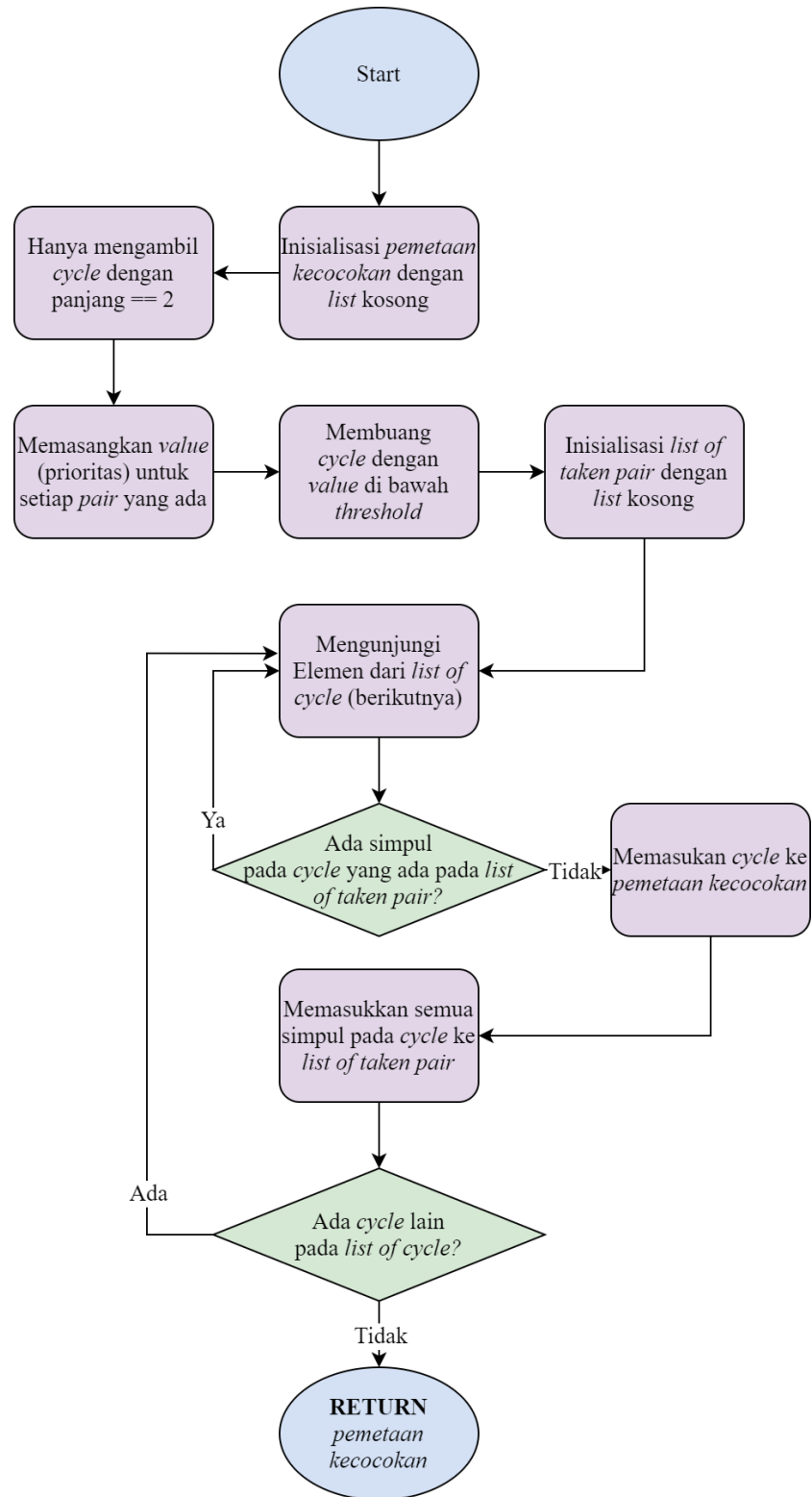
Algoritma pencarian pemetaan kecocokan adalah algoritma yang mencari pemetaan kecocokan paling optimal dari kumpulan suatu entitas menuju entitas-entitas lainnya. Algoritma ini marak digunakan untuk mencari pemetaan kecocokan pengguna-pengguna aplikasi kencan.

Pada kasus ini, algoritma pencarian pemetaan kecocokan digunakan untuk mencari setiap pemetaan kecocokan yang mungkin yang terdapat pada kumpulan pasangan donor-pasien yang tidak cocok (Raja dkk., 2011). Kecocokan adalah saat suatu pasangan donor-resipien dapat memperoleh dan memberikan ginjal ke pasangan donor-resipien yang lainnya. Pada kasus ini, kriteria-kriteria cocok yang digunakan hanyalah kecocokan golongan darah dan juga rendah tidaknya PRA (Raja dkk., 2011). Kriteria seperti uji silang dan dan juga serologi dapat dilakukan setelah kecocokan didapatkan. Apabila hasil uji setelah pencocokan menyatakan bahwa transplantasi tidak dapat dilakukan, maka pencocokan ulang harus dilakukan untuk pasangan-pasangan donor-pasien yang bersangkutan.

Menggunakan algoritma-algoritma pencarian pemetaan kecocokan, dipastikan bahwa setiap pasangan yang mendapatkan kecocokan hanya akan melakukan pertukaran sebanyak satu kali, yang artinya mendapatkan tepat satu buah ginjal dari pendonor yang ada pada pasangan lain, dan juga akan mendonorkan tepat satu buah ginjal pendonor ke pasangan lain. Algoritma pencarian yang sudah ada yang telah diimplementasikan dan marak dipakai pada program KPD adalah *Edmond's Algorithm* dan algoritma *First-accept Match Heuristic*.

II.4.1 Edmond's Algorithm

Algoritma ini merupakan algoritma pencarian untuk kecocokan dua arah (*two-way exchange*). Algoritma ini dijalankan menggunakan sistem prioritas dan algoritma akan menghilangkan semua kemungkinan kecocokan dengan prioritas rendah agar kecocokan dengan prioritas tinggi difokuskan terlebih dahulu (Raja dkk., 2011). *Flowchart* untuk *Edmond's Algorithm* dapat dilihat pada Gambar II.4.1.

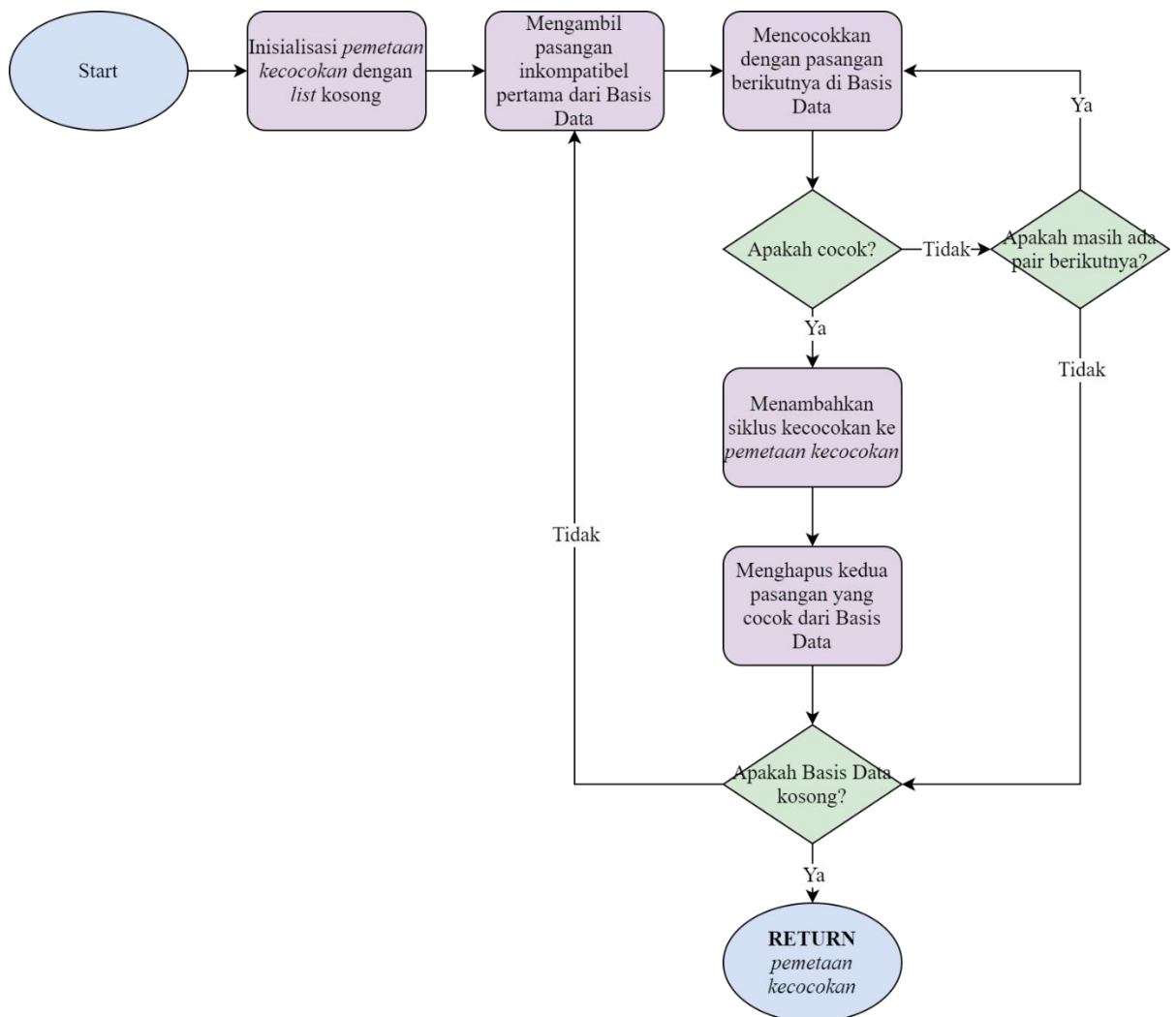


Gambar II.4.1 Flowchart Edmond's Algorithm

II.4.2 Algoritma *First-accept Match Heuristic*

Algoritma ini juga merupakan algoritma pencarian untuk kecocokan dua arah (*two-way exchange*). Algoritma ini dijalankan menggunakan fungsi *heuristic* dimana siklus-siklus dengan kemunculan pertama yang berisikan simpul-simpul yang belum mendapatkan kecocokan sebelumnya akan dikembalikan (Raja dkk., 2011).

Flowchart untuk *First-accept Match Heuristic* dapat dilihat pada Gambar II.4.2.



Gambar II.4.2 Flowchart Algoritma *First-accept Match Heuristic*

II.5 Metrik Kinerja

Metrik adalah suatu standar pengukuran dari kinerja suatu algoritma atau model. Metrik bersifat terukur dan digunakan untuk melihat dan menilai peningkatan kinerja dari suatu algoritma (Tullis & Albert, 2013). Terdapat dua buah metrik kinerja yang digunakan pada tugas akhir ini yaitu:

1. Metrik pertama yang digunakan pada tugas akhir ini adalah *Matching Efficiency* atau efisiensi pencocokan yang merupakan metrik yang digunakan untuk menilai kinerja dari setiap algoritma pencarian pemetaan kecocokan yang dipakai. Efisiensi pencocokan pada program pertukaran ginjal adalah persentase jumlah pasangan donor-pasien yang mendapatkan kecocokan dengan pasangan lain dan dapat melakukan pertukaran dibagi dengan jumlah total pasangan donor-pasien (Raja dkk., 2011). Sebagai suatu contoh, jika ada 51 pasangan yang mendapatkan pertukaran dari 100 pasangan, maka efisiensi pencocokannya adalah 51%.
2. Metrik kedua yang digunakan pada tugas akhir ini adalah waktu eksekusi algoritma. Waktu eksekusi algoritma sendiri merupakan durasi yang mengukur lamanya eksekusi dari suatu potongan kode. Waktu eksekusi algoritma pada umumnya diukur menggunakan satuan milisekon ($1 \text{ milisekon} = 10^{-3} \text{ detik}$).

BAB III

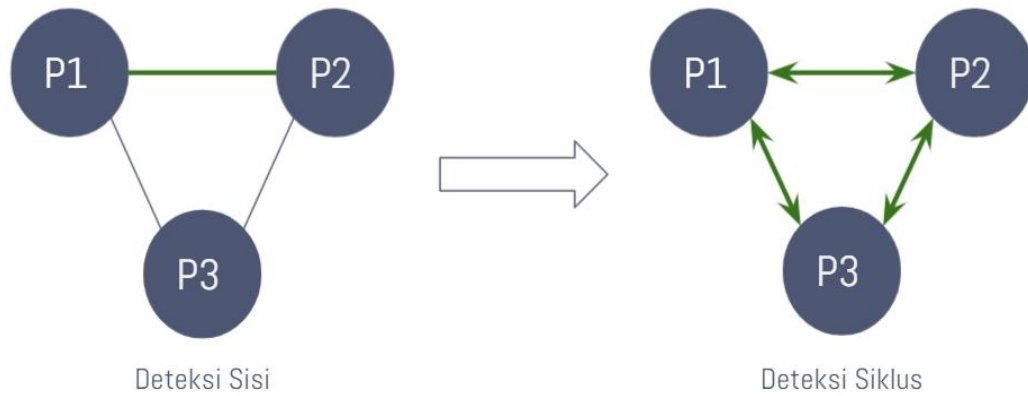
ALGORITMA PENCARIAN N ARAH UNTUK PEMETAAN PERTUKARAN PADA PROGRAM PERTUKARAN GINJAL

Pada bab ini, dijelaskan mengenai analisis yang dilakukan pada algoritma pencarian pemetaan kecocokan yang sudah ada dan rancangan dari solusi yang pada akhirnya diimplementasikan. Bab ini terdiri dari dua subbab, yaitu subbab mengenai masalah yang ada pada algoritma pencarian pemetaan kecocokan dua arah, dan subbab yang mendeskripsikan solusi yang akan diimplementasikan.

III.1 Masalah pada Algoritma Pencarian Pemetaan Kecocokan Dua Arah

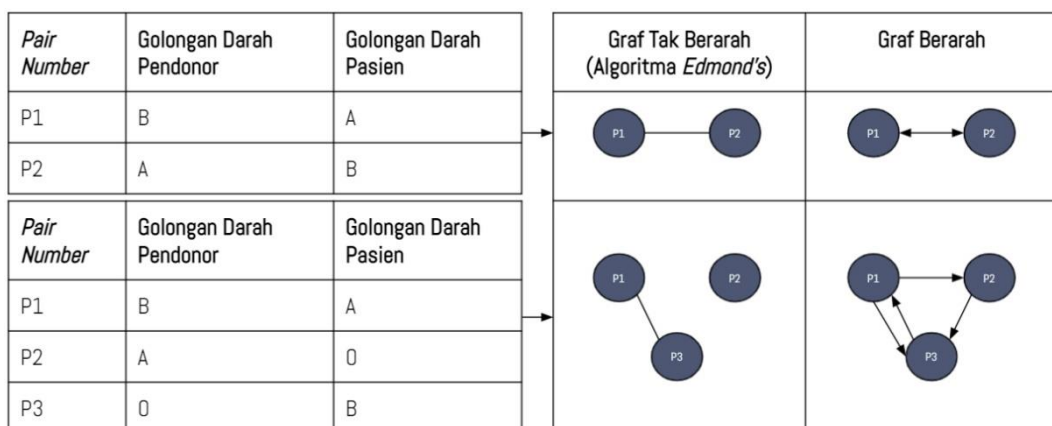
Seperti yang telah diuraikan pada rumusan masalah, *Edmond's algorithm*, algoritma yang sudah ada terkait pencarian solusi pencarian pemetaan pertukaran hanya terbatas untuk pertukaran dua arah yang menutup kemungkinan untuk adanya kecocokan-kecocokan tiga arah atau kecocokan-kecocokan dengan jumlah arah yang lebih banyak, alhasil membatasi jumlah kecocokan yang didapat.

Jika dibedah secara mendalam, *Edmond's algorithm* hanya menggunakan siklus-siklus dua arah, yang didapatkan dari sisi-sisi yang ada pada graf kompatibilitas yang digunakan. Graf kompatibilitas yang dipakai pada *Edmond's algorithm* adalah graf tak berarah. Hal ini dilakukan karena setiap sisi yang ada pada graf menyatakan hubungan bolak-balik antara dua simpul, yang artinya suatu siklus kecocokan dua arah. Sehingga algoritma deteksi siklus yang digunakan pada *Edmond's Algorithm* adalah fungsi untuk melakukan deteksi sisi. Hal ini membatasi adanya kecocokan yang memiliki arah lebih dari dua. Ilustrasi dari keterbatasan ini dapat dilihat pada Gambar III.1.1.



Gambar III.1.1 Deteksi Sisi dibandingkan dengan Deteksi Siklus. Dapat terlihat pada kasus ini bahwa dengan menggunakan algoritma Deteksi Siklus, dapat tercakup tiga pasangan sehingga mampu menyelamatkan tiga pasien dibandingkan hanya dua pada Deteksi Sisi.

Jika *Edmond's algorithm* dikembangkan agar dapat menggunakan siklus, maka kecocokan tiga arah atau lebih dapat tercatat. Namun untuk melakukan deteksi siklus pada graf tak berarah tidaklah optimal. Sisi-sisi dari graf tak berarah yang bersifat bolak-balik merupakan suatu batasan karena menutup keberadaan hubungan satu arah antarpasangan. Dengan tidak adanya hubungan satu arah antarpasangan, siklus-siklus yang tercatat dari suatu graf akan lebih sedikit. Ilustrasi dari pergantian graf ini dapat dilihat pada Gambar III.1.2.



Gambar III.1.2 Penggunaan Graf Tak Berarah dibandingkan dengan Penggunaan Graf Berarah

Dapat dilihat pada Gambar III.1.2. Saat digunakan set data kedua, jika digunakan graf tak berarah, sisi yang terbentuk hanya ada diantara P1 dan P3, sementara jika digunakan graf berarah, siklus dapat terbentuk karena adanya sisi-sisi dari P1 ke P2 ke P3 yang kembali lagi ke P1. Hal ini secara tidak langsung membuktikan dengan penggunaan graf berarah, siklus-siklus yang tidak bersifat bolak-balik dapat terdeteksi sehingga memperbanyak siklus yang dapat diperoleh dari set data yang sama.

Pada algoritma pencarian pemetaan kecocokan secara N arah, tidak hanya menggunakan algoritma deteksi siklus untuk mencari kecocokan-kecocokan secara N arah, struktur data graf kompatibilitas yang digunakan juga diubah untuk menggunakan graf berarah, dengan sisi yang menyatakan hubungan satu arah antara dua simpul pasangan. Pengubahan struktur data graf yang dipakai ini dapat memperbanyak jumlah sisi yang terbentuk pada graf kompatibilitas dan niscaya dapat terbentuk siklus-siklus yang melibatkan lebih banyak pasangan dan alhasil lebih banyak pasien dapat memperoleh donor ginjal.

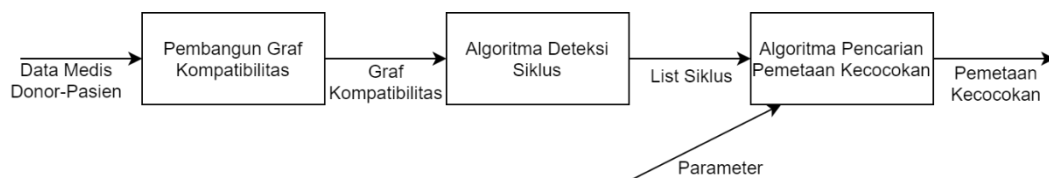
III.2 Deskripsi Solusi

Untuk mengatasi masalah yang telah dijelaskan pada subbab III.1, algoritma pencarian pemetaan kecocokan N arah akan dibuat dengan adanya penggunaan algoritma deteksi siklus dan juga pemakaian graf kompatibilitas yang berarah yang dapat menyatakan arah kemungkinan suatu donor dari suatu pasangan dapat mendonorkan ginjalnya kepada pasien dari pasangan lain. Hal ini akan meningkatkan jumlah sisi yang ada pada graf dan niscaya mempermudah pencarian untuk pemetaan kecocokan bersifat N arah. Dengan karakteristik graf yang telah dijelaskan pada subbab II.3, dapat dibuat algoritma-algoritma pencarian pemetaan kecocokan yang mampu mencari pemetaan kecocokan secara N arah.

Secara konseptual, solusi lengkap yang dipilih adalah algoritma-algoritma pencarian pemetaan kecocokan yang bekerja secara N arah. Untuk memperjelas rancangan solusi yang dibuat, telah dibuat suatu *Block Diagram*. Diagram yang dibuat merepresentasikan alur jalannya data dan proses-proses yang dilakukan

untuk mendapatkan pemetaan kecocokan. *Block Diagram* dari solusi yang akan diimplementasikan dapat dilihat pada Gambar III.2.1.

Proses-proses pada Gambar III.2.1 terbagi menjadi tiga, yaitu pembangun graf kompatibilitas, algoritma deteksi siklus, dan juga algoritma pencarian pemetaan kecocokannya itu sendiri. Pengguna memasukkan data medis donor-pasien untuk dapat membangun graf kompatibilitas, dari graf kompatibilitas yang terbuat dari data, dapat dideteksi siklus-siklus yang ada, menggunakan *list* siklus yang terbentuk pada graf kompatibilitas, dan juga nilai-nilai untuk parameter-parameter yang dibutuhkan, algoritma pencarian pemetaan kecocokan akan berjalan dan menghasilkan pemetaan kecocokan.



Gambar III.2.1 *Block Diagram* Rancangan Solusi

III.2.1 Data Medis Donor-Pasien

Data medis donor dan pasien adalah data terstruktur yang berisikan nomor dari pasangan, golongan darah pendonor ginjal, golongan darah pasien calon penerima donor ginjal, dan juga *Panel Reactive Antibody* (PRA) dari pasien calon penerima transplantasi ginjal.

Nomor dari pasangan merupakan *primary key* dari tabel data ini. Nomor pasangan memiliki format “Pxxx” dimana P artinya suatu nomor tersebut adalah sebuah pasangan ketidakcocokan, dan xxx adalah nomor dari pasangan itu sendiri. Data ini memiliki struktur seperti yang dapat dilihat pada Tabel III.3.1.

Tabel III.2.1 Struktur Data Medis Donor dan Pasien yang Dibutuhkan

Atribut		Keterangan
PK	pair_num	Nomor unik penanda pasangan donor dan resipien
	donor_bloodtype	Golongan Darah Pendonor
	recipient_bloodtype	Golongan Darah Resipien
	pra	<i>Panel-Reactive Antibody</i> dari pasangan

Tabel ini berisikan data dari pasangan yang mendaftar dari berbagai tanggal. Format tanggal yang digunakan ada pada nama tabel, yaitu “dr%YYYY_%mm_%dd”, dengan %YYYY adalah tahun, %mm adalah bulan, dan %dd adalah hari. Setiap pendaftar yang mendaftarkan diri pada hari tersebut akan memiliki satu tabel tersendiri. Contoh dari data medis donor-pasien tercantum pada Tabel III.2.2.

Tabel III.2.2 Contoh Data Medis Donor-Pasien

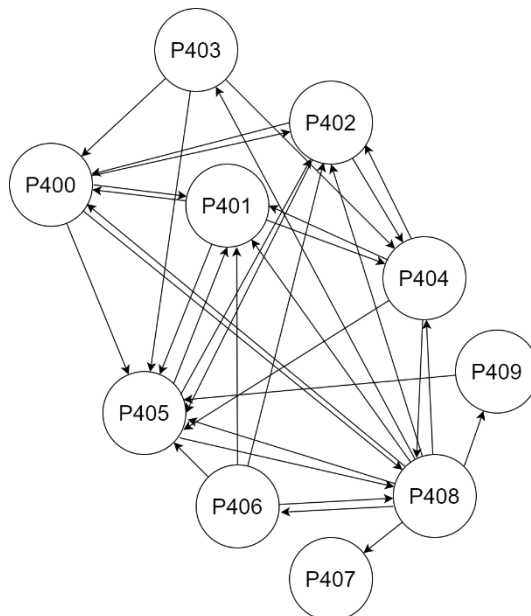
pair_num	donor_bloodtype	recipient_bloodtype	PRA
P400	A	B	5
P401	B	A	92
P402	B	A	5
P403	B	O	10
P404	A	B	5
P405	A	AB	28
P406	A	O	28
P407	A	O	28
P408	O	A	90
P409	AB	O	18

III.2.2 Pembangun Graf Kompatibilitas (*Compatibility Graph*)

Sebelum data medis pasangan donor-pasien dimasukkan ke dalam algoritma pencarian pemetaan kecocokan, data medis yang merupakan data terstruktur berisikan informasi medis donor dan pasien diubah terlebih dahulu menjadi suatu graf kompatibilitas. Paling sedikit, data medis donor dan pasien mengandung golongan darah pendonor, golongan darah pasien, dan juga tanda pengenal unik untuk setiap pasangan.

Graf kompatibilitas yang dibuat merupakan suatu graf berarah dengan simpul yang merupakan pasangan donor-pasien yang tidak cocok dan sisi yang merupakan arah donasi yang mungkin dari suatu pasangan ke pasangan lainnya. Untuk implementasinya, akan dibuat suatu kelas yang memiliki atribut berupa *adjacency list*, list siklus, dan juga untuk penyimpanan data masukannya. Data yang dimasukkan akan langsung dikonversi menjadi *adjacency list* yang dibuat dengan cara membandingkan setiap pair data dengan pair lainnya untuk membuat sisi di antara dua pair yang bersangkutan dengan sifat *many-to-many*.

Sebagai contoh, dari data yang tercantum pada Tabel III.2.2, dapat terbentuk Graf Kompatibilitas yang dapat dilihat pada Gambar III.2.2.

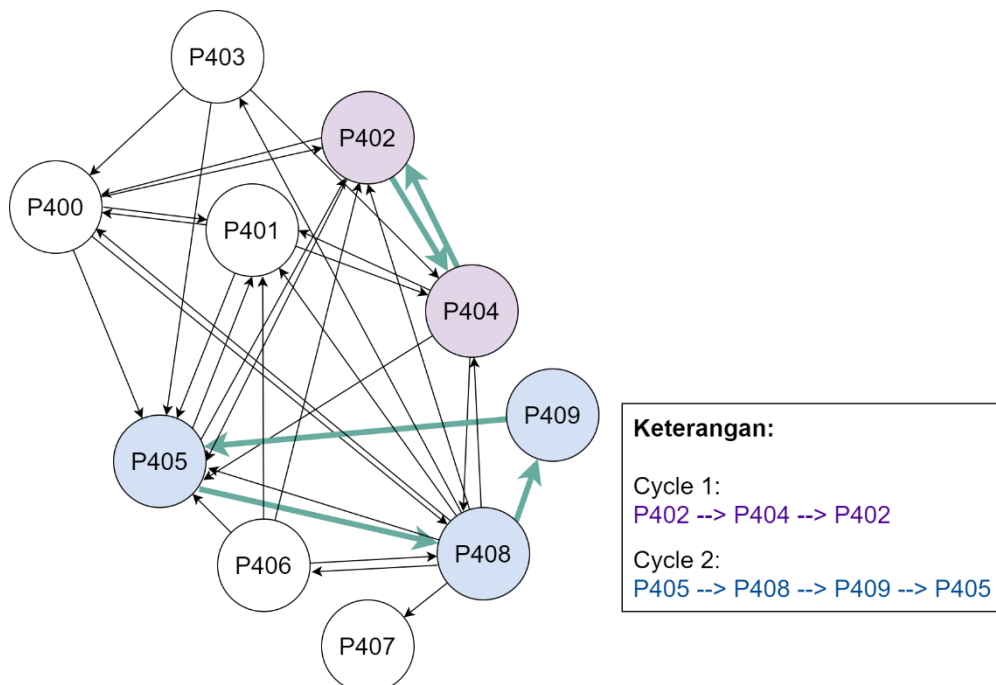


Gambar III.2.2 Contoh Graf Kompatibilitas untuk Data Contoh

III.2.3 Algoritma Deteksi Siklus (Cycle Detection)

Setelah graf kompatibilitas terbentuk, perlu dicari kemungkinan-kemungkinan pemetaan kecocokan yang terdapat pada graf. Untuk melakukan ini, dipakai algoritma deteksi siklus. Algoritma deteksi siklus pada umumnya digunakan untuk melakukan pengecekan apakah suatu graf memiliki siklus atau tidak. Namun pada kasus ini, algoritma deteksi siklus pada graf berarah digunakan untuk mencari keberadaan semua jalur pergerakan pada graf yang dimulai dan selesai pada simpul yang sama. Pendekatan yang digunakan dalam implementasi ini adalah untuk melakukan deteksi berulang kali dimulai dari setiap simpul.

Pada kasus ini, siklus yang terbentuk merupakan pemetaan kecocokan yang menandakan pertukaran ginjal yang mungkin untuk dilakukan karena terjadi kecocokan. Penjelasan algoritma deteksi siklus dicantumkan pada subbab II.3.1. Algoritma deteksi siklus merupakan suatu algoritma *Depth First Search* (DFS). Sebagai contoh, dari graf kompatibilitas pada Gambar III.2.2, dapat diperoleh beberapa buah siklus seperti yang dapat dilihat pada Gambar III.2.3.



Gambar III.2.3 Beberapa Siklus yang didapatkan dari Graf Kompatibilitas Contoh

III.2.4 Algoritma Pencarian Pemetaan Kecocokan

Dengan adanya *list* siklus-siklus di dalam graf kompatibilitas yang terbentuk dari banyaknya data pasangan donor dan pasien berkebutuhan ginjal yang tidak cocok, algoritma pencarian pemetaan kecocokan akan mencari suatu pemetaan kecocokan yang adalah sebuah subgraf yang berisi kumpulan siklus yang melibatkan paling banyak simpul. Setiap simpul hanya boleh berada dalam satu buah siklus, yang berarti setiap pasangan hanya akan menerima dan memberikan tepat satu buah ginjal. Representasi pemrograman dari pemetaan kecocokan adalah *list* siklus.

Telah dibuat dua jenis pendekatan untuk algoritma-algoritma pencarian pemetaan kecocokan yaitu:

1. Algoritma *First-accept Searching*
2. Algoritma *Priority-based Searching*

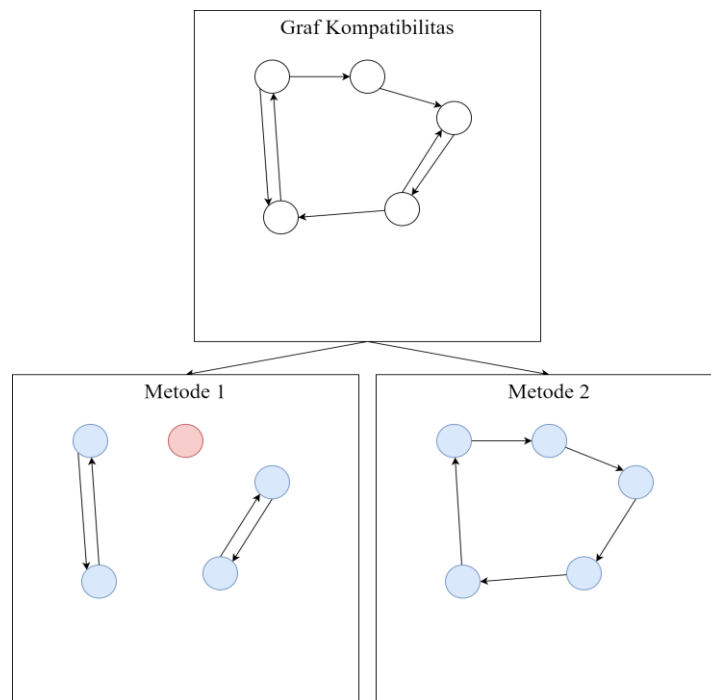
Algoritma-algoritma ini membutuhkan beberapa parameter. Parameter utama yang digunakan adalah nilai N yang merupakan jumlah arah pencocokan yang dapat terjadi.

Terdapat dua buah metode untuk menggunakan nilai N . Metode pertama adalah metode dimana nilai N digunakan sebagai nilai maksimum pertukaran yang terjadi. Metode kedua adalah metode dimana nilai N digunakan sebagai panjang eksak pertukaran yang terjadi. Pada metode pertama, siklus kecocokan yang didapatkan oleh sistem dapat berupa pertukaran sebanyak dua arah hingga sebanyak N arah. Sementara pada metode kedua, siklus kecocokan yang didapatkan oleh sistem sudah pasti berupa pertukaran sebanyak N arah. Jika N yang dipakai adalah 5 saat metode yang dipilih adalah metode maksimum, maka pertukaran dua arah, pertukaran tiga arah, pertukaran empat arah, dan juga pertukaran lima arah yang masih mungkin diambil tetap akan diambil. Jika metode yang dipilih adalah metode eksak, maka hanya pertukaran lima arah saja yang akan diambil.

Meskipun metode pertama seolah-olah terdengar seperti metode yang sudah pasti akan menghasilkan jumlah kecocokan terbanyak, hal ini tidak selalu benar. Karena graf berarah bersifat *many-to-many*, suatu simpul dapat berada pada lebih dari satu

buah siklus pada siklus-siklus yang didapatkan dari graf yang bersangkutan. Sehingga ada kasus dimana jumlah kecocokan yang didapat lebih banyak apabila metode yang dipilih adalah metode kedua.

Contoh kasus dimana metode kedua unggul dapat dilihat pada Gambar III.2.4. Kasus yang dicantumkan pada Gambar III.2.4 adalah kasus saat N bernilai 5. Pada kasus ini, saat menggunakan metode pertama, siklus-siklus dua arah yang terjadi pada graf kompatibilitas dapat diambil terlebih dahulu, membuat siklus lima arah yang terjadi tidak terambil, alhasil mengurangi jumlah pasangan yang mendapatkan kecocokan. Sementara itu, saat menggunakan metode kedua, siklus dengan Panjang lima arah akan diambil karena metode kedua mengambil semua siklus dengan panjang yang sama dengan nilai N .



Gambar III.2.4 Kasus dimana metode eksak unggul dibandingkan metode maksimum

Untuk algoritma berbasis prioritas seperti *Priority-based Searching*, selain metode penggunaan nilai N , ada pula satu parameter penting untuk penentuan jenis prioritas seperti apa yang dipakai. Prioritas jenis pertama bekerja dengan

memprioritaskan siklus-siklus terpanjang dari list siklus. Implikasi dari prioritas jenis pertama ini adalah dengan lebih banyak mengambil siklus-siklus yang panjang, lebih banyak kecocokan yang didapatkan sehingga lebih banyak pasien yang terselamatkan. Namun saat lebih banyak pasien yang terselamatkan, terdapat kemungkinan bahwa pasien-pasien yang memiliki golongan darah langka dan sulit mendapatkan kecocokan untuk tertinggal dan tidak terselamatkan. Oleh sebab itu dibuatlah metode penentuan prioritas jenis kedua. Metode penentuan prioritas jenis kedua bekerja dengan memprioritaskan siklus-siklus yang terdiri dari pasangan-pasangan pendonor-pasien yang memiliki kemungkinan kecocokan tersedikit, yang implikasi pada graf kompatibilitasnya adalah simpul-simpul yang memiliki jumlah sisi tersedikit. Dengan mengimplementasikan prioritas jenis kedua ini, meskipun kemungkinan untuk mendapatkan banyak kecocokan dapat menurun, namun lebih banyak pasien ber kondisi jarang yang dapat diselamatkan.

Jika dibandingkan dengan algoritma pencarian pemetaan kecocokan dua arah, algoritma pencarian pemetaan kecocokan N arah pada umumnya memiliki suatu kelemahan. Kelemahan ini adalah tantangan dimana siklus-siklus pada graf kompatibilitas harus dapat diperoleh sebelum algoritma ini dapat dipanggil. Selain itu, dengan menggunakan algoritma pencarian pemetaan kecocokan N arah, rumah sakit yang bersangkutan perlu mampu menangani operasi yang berjalan secara paralel dan bersamaan untuk setiap pencocokannya. Kedua buah algoritma pencarian pemetaan kecocokan N arah yang dibuat yang masing-masing akan dijelaskan pada subbab III.2.4.1 dan III.2.4.2.

III.2.4.1 Algoritma *First-accept Searching*

Algoritma ini mencari pemetaan kecocokan secara *first-accept*, yang berarti siklus-siklus kemungkinan pemetaan kecocokan yang pertama didapatkan akan dikembalikan terlebih dahulu. Algoritma ini dapat memastikan tidak adanya pasangan yang mendapatkan kecocokan dua kali dengan menghilangkan siklus-siklus yang mengandung pasangan yang telah ada pada siklus-siklus yang telah dikembalikan sebelumnya.

Algoritma ini menggunakan dua buah parameter, yaitu nilai N dan juga metode pemakaian nilai N . Seperti yang telah dijelaskan pada subbab III.2, Metode-metode yang ada adalah metode penggunaan nilai N sebagai panjang pertukaran maksimum dan metode penggunaan nilai N sebagai panjang pertukaran eksak. *Flowchart* untuk algoritma *First-accept Searching* dapat dilihat pada Gambar III.2.5.

Inspirasi dari algoritma ini diambil dari *paper* rujukan utama bernama *First-accept Match Heuristic*. Hal yang dimodifikasi dari algoritma ini adalah algoritma tidak berjalan menggunakan setiap pair yang lalu dihubungkan dengan pair-pair lain dengan sisi, tetapi menggunakan *list* siklus. Kompleksitas waktu dari algoritma ini adalah $O(n^2)$.

III.2.4.2 Algoritma *Priority-based Searching*

Tidak seperti algoritma *first-accept searching* yang dinyatakan pada subbab III.2.4.1, algoritma ini akan mencari pemetaan kecocokan dari list siklus dengan skala prioritas. Siklus-siklus yang memiliki prioritas lebih tinggi akan dikembalikan terlebih dahulu dibandingkan siklus-siklus dengan prioritas yang lebih rendah.

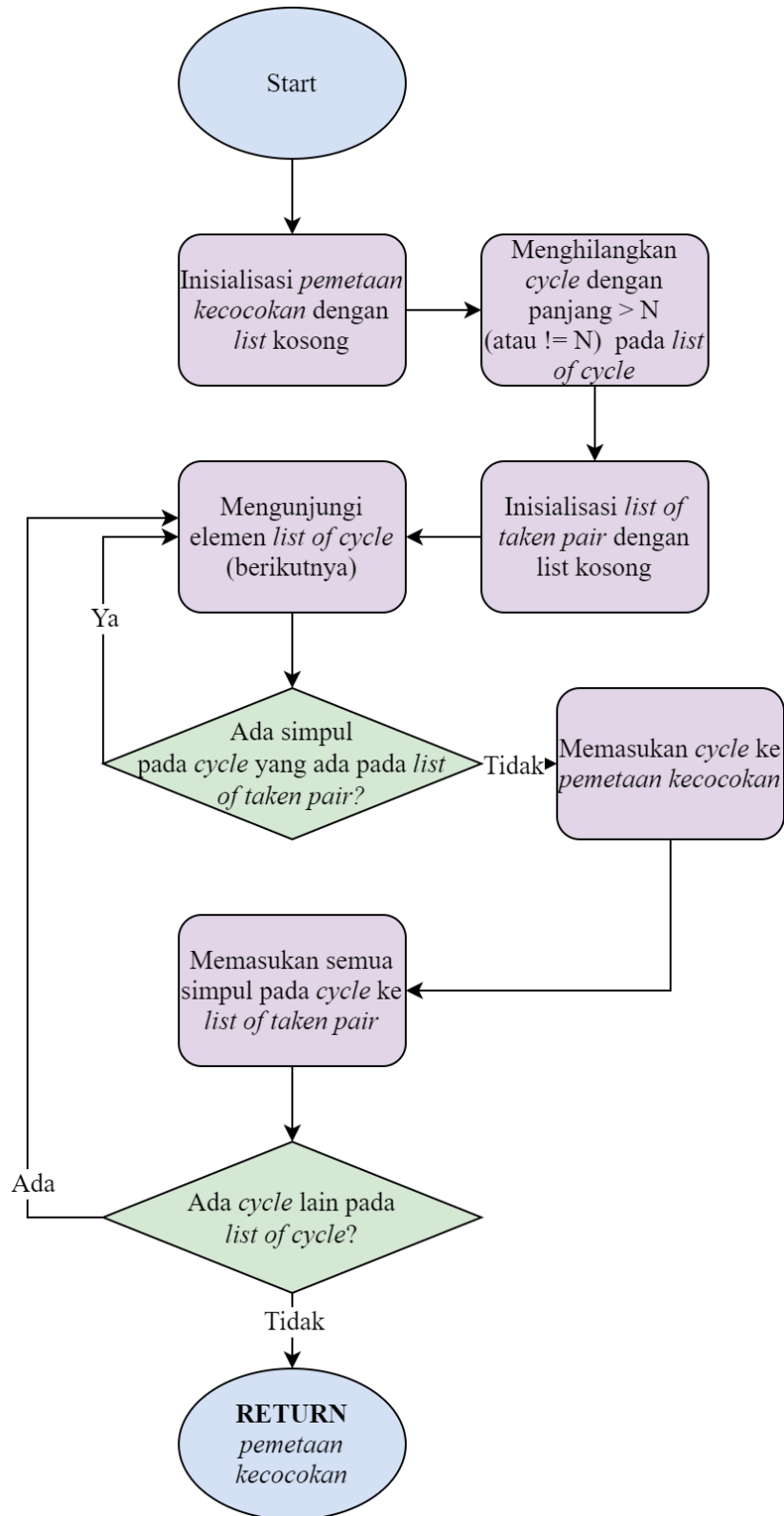
Pada algoritma ini, parameter-parameter yang diperlukan adalah nilai N , metode pemakaian nilai N , dan juga metode pemilihan prioritas untuk setiap siklus kemungkinan kecocokannya. Seperti Algoritma *First-Accept Searching*, metode penggunaan nilai N dapat dipilih antara sebagai panjang pertukaran maksimum atau sebagai panjang pertukaran eksak. Untuk pemilihan prioritas, dapat dipilih dua jenis prioritas.

Seperti yang telah dijelaskan pada subbab III.2, pendekatan pertama adalah penentuan prioritas dengan cara *greedy*, yaitu untuk memberikan prioritas yang lebih tinggi untuk siklus-siklus yang lebih panjang dibandingkan siklus-siklus yang lainnya, dengan harapan dengan lebih banyaknya siklus berorde tinggi, lebih banyak pasangan yang akan mendapatkan kecocokan. Tidak seperti pendekatan pertama, pendekatan pemilihan prioritas kedua yang dinamakan dengan metode penentuan prioritas *infrequent*, memberikan prioritas yang lebih tinggi untuk siklus-siklus yang terdiri dari pasangan-pasangan yang memiliki kemungkinan kecocokan yang rendah. Kemungkinan kecocokan yang rendah pada implementasinya

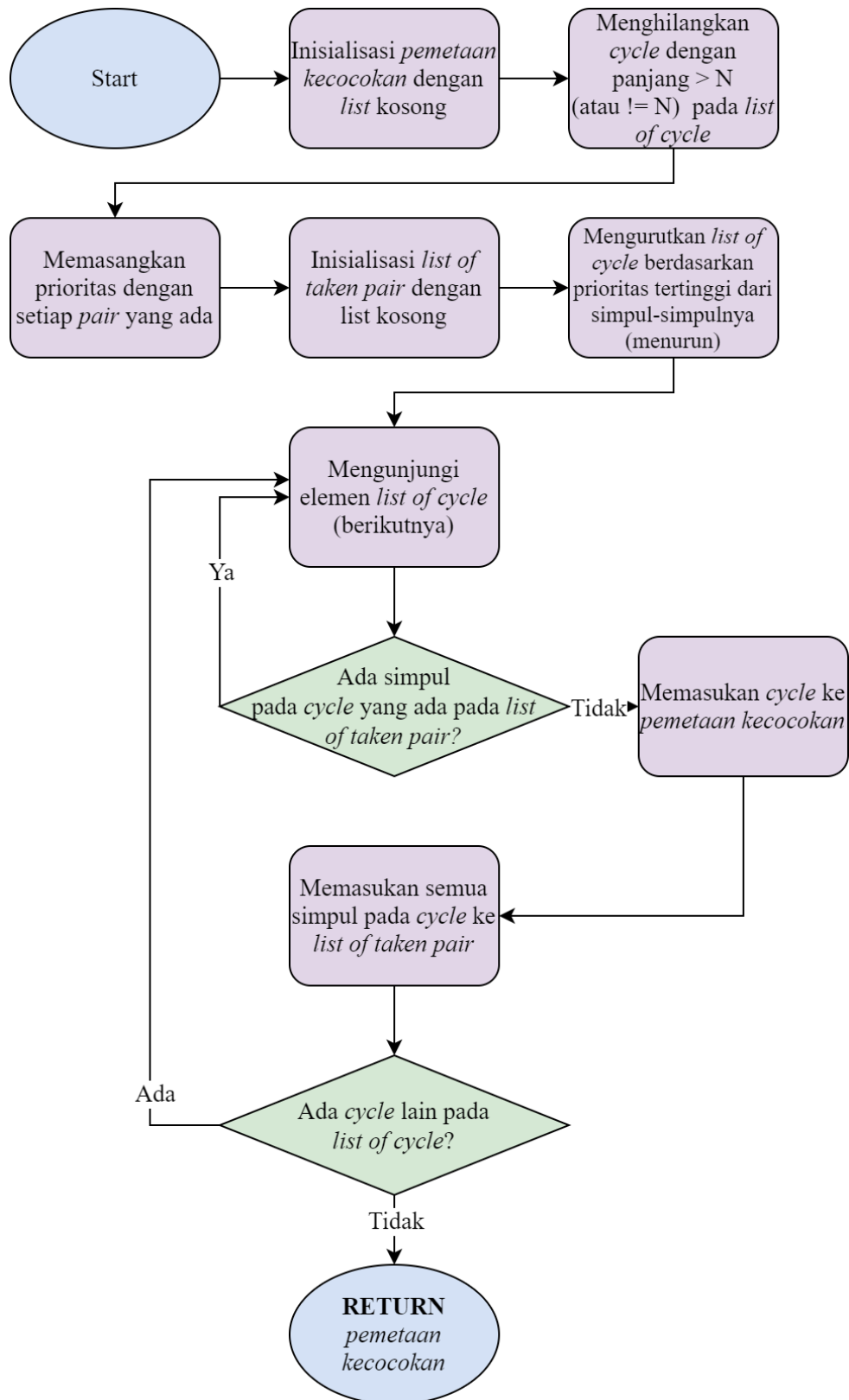
mengartikan bahwa simpul pada graf memiliki sisi yang lebih sedikit dibandingkan simpul-simpul lainnya. *Flowchart* dari algoritma *priority-based searching* dapat dilihat pada Gambar III.2.6.

Inspirasi dari algoritma ini adalah algoritma yang sudah ada, *Edmond's Algorithm*. Hanya saja algoritma *priority-based* ini tidak menggunakan parameter *priority threshold* yang digunakan *Edmond's Algorithm* yang berfungsi untuk mengeliminasi siklus-siklus dengan prioritas di bawah *threshold*. Selain tidak mengeliminasi siklus dengan prioritas rendah, algoritma ini juga menggunakan pendekatan prioritas yang berbeda. Penentuan prioritas dengan metode *greedy* tidak memiliki inspirasi dari manapun. Sementara untuk penentuan prioritas dengan metode *infrequent* digunakan untuk mendapatkan kecocokan untuk pasangan-pasangan yang jarang dan bersifat langka seperti pasangan dengan pendonor bergolongan darah AB. Pasangan-pasangan yang bersifat jarang ini mendapatkan kerugian dan lebih sulit mendapatkan ginjal dibandingkan pasangan-pasangan yang lebih sering ada.

Kompleksitas dari algoritma ini adalah $O(n^2)$. Namun untuk pendekatan pemilihan prioritas *infrequent*, karena perlu dilakukan *lookup* menuju graf kompatibilitas untuk mendapatkan jumlah sisi per simpul di dalam suatu siklus, algoritma akan berjalan relatif lebih lama dibandingkan saat pendekatan pemilihan prioritas yang digunakan adalah *greedy*. *Lookup* ini sendiri memiliki kompleksitas $O(n^2)$.



Gambar III.2.5 Flowchart Algoritma First-Accept Searching



Gambar III.2.6 Flowchart Algoritma Priority-based Searching

BAB IV

IMPLEMENTASI DAN EVALUASI

Pada bab ini, dijelaskan mengenai implementasi dan juga evaluasi yang dilakukan. Terdapat tiga buah subbab besar pada bab ini. Subbab untuk menjelaskan mengenai implementasi yang dilakukan, subbab mengenai metode dan juga lingkungan pengujian, dan diakhiri dengan subbab mengenai hasil-hasil eksperimen.

IV.1 Implementasi

Implementasi dilakukan secara terurut berdasarkan urutan dari *flow* data pada diagram blok pada Gambar III.2.1. Pembangun graf kompatibilitas, Algoritma deteksi siklus, dan diakhiri dengan algoritma pencarian pemetaan kecocokan. Implementasi dilakukan menggunakan Bahasa pemrograman Python v3.8.5. Kode sumber lengkap dari pengerjaan keseluruhan tugas akhir ini dapat diakses pada pranala <https://github.com/Mingtaros/Kidney-Exchange-Match-Mapping-System>.

IV.1.1 Pembangun Graf Kompatibilitas

Berdasarkan data medis donor-pasien yang telah dispesifikasikan pada subbab III.2.1 dan juga penjelasan mengenai pembangun graf kompatibilitas pada subbab III.2.2, dibuatlah suatu algoritma untuk membentuk *Directed Graph* kecocokan berdasarkan data yang digunakan. Untuk setiap pasangan pada data, akan dipasang-pasangkan dan dicek kecocokannya menggunakan golongan darah pendonor dari pasangan pertama dan juga golongan darah resipien pasangan lainnya. Jika cocok, maka sisi dapat terbentuk.

Representasi dari graf kompatibilitas ini adalah *adjacency list* dengan elemen berbentuk *key-value* pair. Untuk setiap elemen, simpul digunakan sebagai *key* dan simpul-simpul yang dituju oleh simpul *key* melalui sisi digunakan sebagai *value*. Contoh dari *adjacency list* graf kompatibilitas dapat dilihat pada Tabel IV.1.1.

Tabel IV.1.1 Contoh Graf Kompatibilitas dengan Representasi *Adjacency List*

```
{
  "P101": ["P102", "P104"],
  "P102": ["P104"],
  "P103": [],
  "P104": ["P101", "P103"],
}
```

IV.1.2 Algoritma Deteksi Siklus

Menggunakan graf kompatibilitas yang telah dijelaskan pada tahapan implementasi sebelumnya, siklus-siklus yang ada pada graf kompatibilitas dapat dicari menggunakan algoritma deteksi siklus yang dijelaskan pada subbab III.2.3. Algoritma deteksi siklus ini adalah algoritma rekursif yang berfungsi untuk mencari setiap kemungkinan siklus dari setiap simpul yang ada pada graf, terkecuali untuk simpul-simpul yang tidak memiliki sisi ke arah luar. Setiap siklus yang telah terbuat akan disimpan dalam suatu *list* dan saat deteksi siklus dari semua simpul telah selesai dilakukan, *list* dari siklus ini akan dikembalikan.

Representasi dari siklus ini sendiri adalah suatu *list* berisikan simpul-simpul pada graf secara berurutan. Jika tertuliskan ["A", "B", "C"], maka terdapat siklus tiga elemen berbentuk $A \rightarrow B \rightarrow C \rightarrow A$. Dengan representasi siklus berbentuk *list*, maka *list* siklus yang dikembalikan berbentuk sebagai *list of lists*. Contoh dari *list* siklus dapat dilihat pada Tabel IV.1.2.

Tabel IV.1.2 Contoh *List* Siklus dengan Representasi *List of Lists*

```
[
  ["P101", "P102", "P104"],
  ["P101", "P104"],
  ["P103", "P105"]
]
```

IV.1.3 Algoritma Pencarian Pemetaan Kecocokan

Menggunakan *list* siklus hasil keluaran dari tahapan sebelumnya, algoritma pencarian pemetaan kecocokan dapat dijalankan. Implementasi dari algoritma

pencarian pemetaan kecocokan terdiri dari suatu *parent class* bernama ExchangeAlgorithm. *Children class* dari *parent class* ini adalah EdmondsAlgorithm sebagai kelas untuk *Edmond's algorithm*, FirstAcceptNWay sebagai kelas untuk algoritma *First-accept Searching*, dan PriorityBasedNWay sebagai kelas untuk algoritma *Priority based Searching*. *Class Diagram* untuk algoritma-algoritma pencarian pemetaan kecocokan dapat dilihat pada Gambar IV.1.1.

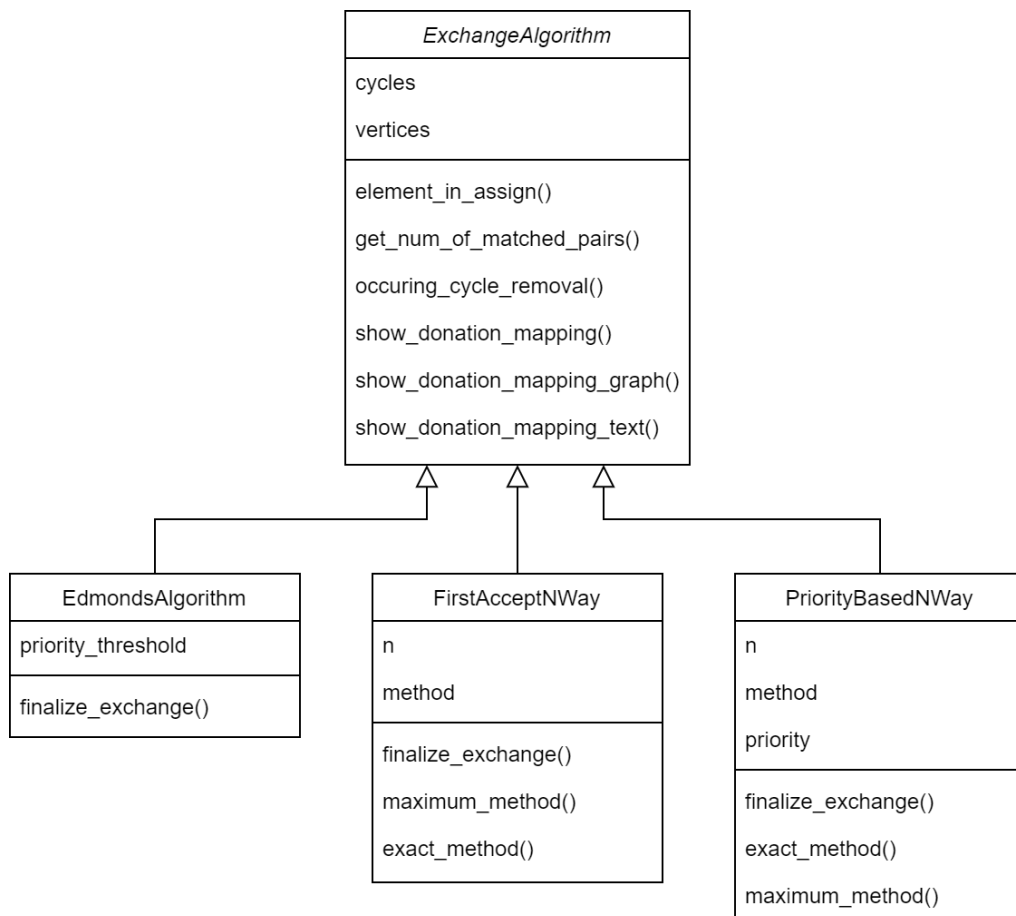
Hasil dari algoritma pencarian pemetaan kecocokan ini adalah *list* siklus masukan sebelumnya yang telah direduksi agar tidak terjadi *overlapping* dari setiap simpul yang mendapatkan kecocokannya. Perlu diperhatikan bahwa pada *list siklus* yang didapatkan, suatu pasangan donor-resipien dapat hadir di dalam lebih dari satu buah siklus sehingga hubungan bersifat *many-to-many* untuk simpul-simpul yang mendapatkan kecocokannya. Algoritma ini dijalankan agar pasangan-pasangan yang mendapatkan kecocokan memiliki hubungan *one-to-one* terhadap pasangan-pasangan lainnya. Contoh keluaran dari tahap ini dapat dilihat pada Tabel IV.1.3. Sementara untuk kode lengkap algoritma-algoritma pencarian pemetaan kecocokan dapat dilihat pada Lampiran A.

Tabel IV.1.3 Contoh Keluaran Algoritma Pencarian Pemetaan Kecocokan

```
[
  ["P101", "P102", "P104"],
  ["P103", "P105"]
]
```

IV.1.4 Aplikasi Pengujian

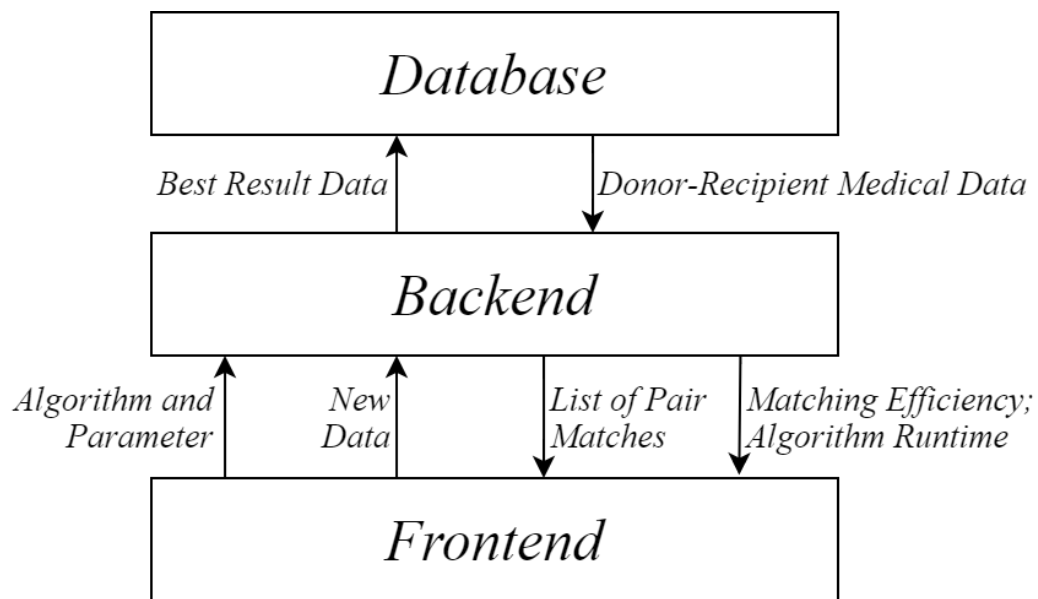
Untuk mempermudah dan mempercepat pengujian, dibuat juga suatu aplikasi sederhana dengan tampilan *dashboard* yang dapat digunakan secara leluasa untuk mengubah-ubah jenis pendekatan algoritma dan juga parameter-parameternya untuk dibandingkan satu dengan yang lainnya sehingga dapat diperoleh solusi optimum lokal.



Gambar IV.1.1 *Class Diagram* untuk Algoritma Pencarian Pemetaan Kecocokan

Optimum lokal pada kasus ini adalah algoritma dengan parameter-parameter tertentu yang berhasil memperoleh hasil kinerja tertinggi dibandingkan algoritma-algoritma lainnya yang diuji pada saat yang bersamaan menggunakan data yang sama. Jumlah dari kombinasi algoritma dan parameter yang dapat diuji pada satu saat dibatasi agar sistem tidak perlu melakukan terlalu banyak kalkulasi sehingga *overload*, atau berjalan terlalu lama dengan *load* yang tinggi. Alasan kedua dari pembatasan ini adalah agar pengguna aplikasi pengujian ini tidak perlu melihat terlalu banyak informasi pada saat yang bersamaan, yang dapat menyebabkan semakin tidak efektifnya *dashboard* yang dibuat.

Setelah didapatkan hasil dari setiap kombinasi yang diujikan, aplikasi akan memperlihatkan tabel perbandingan dan juga *bar plot* yang berisikan hasil performa masing-masing kombinasi untuk setiap metrik yang diujikan. Diagram blok dari aplikasi testing yang dibuat dapat dilihat pada Gambar IV.1.2. Patut dicatat bahwa aplikasi pengujian ini diimplementasi hanya untuk kebutuhan pengujian dan bukan untuk membuat produk jadi.



Gambar IV.1.2 Block Diagram Aplikasi Pengujian

IV.1.4.1 Database

Basis data digunakan untuk menyimpan dua hal, data medis donor dan pasien calon penerima transplantasi ginjal, dan juga data untuk penyimpanan sementara hasil terbaik yang didapatkan dari kumpulan pasangan dari suatu tanggal. Basis data bersifat terstruktur dan tabular. Oleh karena itu, digunakan PostgreSQL 12.6 untuk implementasinya. Data yang dimiliki dimasukkan secara otomatis menggunakan Python baik dari *file-file* yang ada untuk merepresentasikan pasangan ketidakcocokan, maupun data hasil generasi dari algoritma. Untuk mempermudah akses dan juga *create, read, update, delete* (CRUD) basis data, diimplementasikan

juga kelas PostgreSQLHelper menggunakan Python. Kelas PostgreSQLHelper berisikan metode-metode untuk mengeksekusi *query-query* yang bersifat menyunting basis data dan bersifat membaca dari basis data. PostgreSQLHelper yang diimplementasikan menggunakan *library* Python yaitu *psycopg2* 2.8.6.

A. Data Medis Donor-Pasien

Data medis pasangan donor-pasien diambil dari pranala <https://rdm.inesctec.pt/dataset/ii-2020-001>. Struktur dari data ini sesuai dengan struktur data yang ada pada rancangan solusi pada subbab III.3.1.1, dimana data mengandung nomor penanda unik setiap pasangan, golongan darah pendonor, golongan darah resipien, dan juga *Panel-Reactive Antibody* (PRA) yang dihasilkan oleh uji lab pasangan. Distribusi untuk jumlah data golongan darah dapat dilihat pada Tabel IV.1.4.

B. Data Hasil Terbaik

Hasil terbaik yang bisa didapatkan dari suatu kumpulan data pasangan donor dan pasien didapatkan dari nilai optimum lokal yang dipilih berdasarkan kombinasi-kombinasi yang diuji coba untuk data yang bersangkutan. Hasil terbaik ini disimpan pada basis data agar dapat dilakukan komparasi secara tidak langsung dengan kombinasi-kombinasi baru apabila menggunakan data medis donor-pasien yang sama. Karena data medis donor-pasien dipisahkan berdasarkan tanggal pendaftaran, optimum lokal didapatkan untuk setiap data pada suatu tanggal pendaftaran. Struktur untuk penyimpanan hasil terbaik dapat dilihat pada Tabel IV.1.5.

IV.1.4.2 Backend

Backend dari aplikasi testing diimplementasikan menggunakan bahasa pemrograman Python 3.8.5 dengan kerangka kerja utama Django. Terdapat beberapa buah *endpoint* yang diimplementasikan. Penjelasan singkat mengenai fungsi-fungsi dari masing-masing *endpoint* yang penting adalah sebagai berikut:

- Fungsi untuk mendapatkan hasil pemetaan kecocokan saat diberikan spesifikasi algoritma pencocokan dan juga data yang dipilih.

- Fungsi untuk mendapatkan list dari semua tanggal yang ada minimal satu buah pendaftar.
- Fungsi untuk mendapatkan data lengkap dari basis data berdasarkan tanggal dan nomor-nomor pasangan yang diminta.
- Fungsi untuk menyimpan hasil pencarian pemetaan kecocokan terbaik ke basis data
- Fungsi untuk mengambil hasil pencarian pemetaan kecocokan terbaik dari basis data
- Fungsi untuk menginput data pada tanggal pendaftaran menuju basis data.

Tabel IV.1.4 Distribusi Data Medis Donor dan Pasien

Golongan Darah Pendonor	Golongan Darah Pasien	Jumlah
A	A	142
	B	314
	O	945
	AB	19
B	A	310
	B	29
	O	390
	AB	10
O	A	187
	B	81
	O	275
	AB	39
AB	A	104
	B	39
	O	117
	AB	0

Tabel IV.1.5 Struktur Data Hasil Terbaik pada Basis Data

Atribut		Keterangan
PK	date	Tanggal Pendaftaran
	matched_pairs	Pemetaan kecocokan dengan representasi <i>list</i> siklus yang merupakan Hasil terbaik yang disimpan

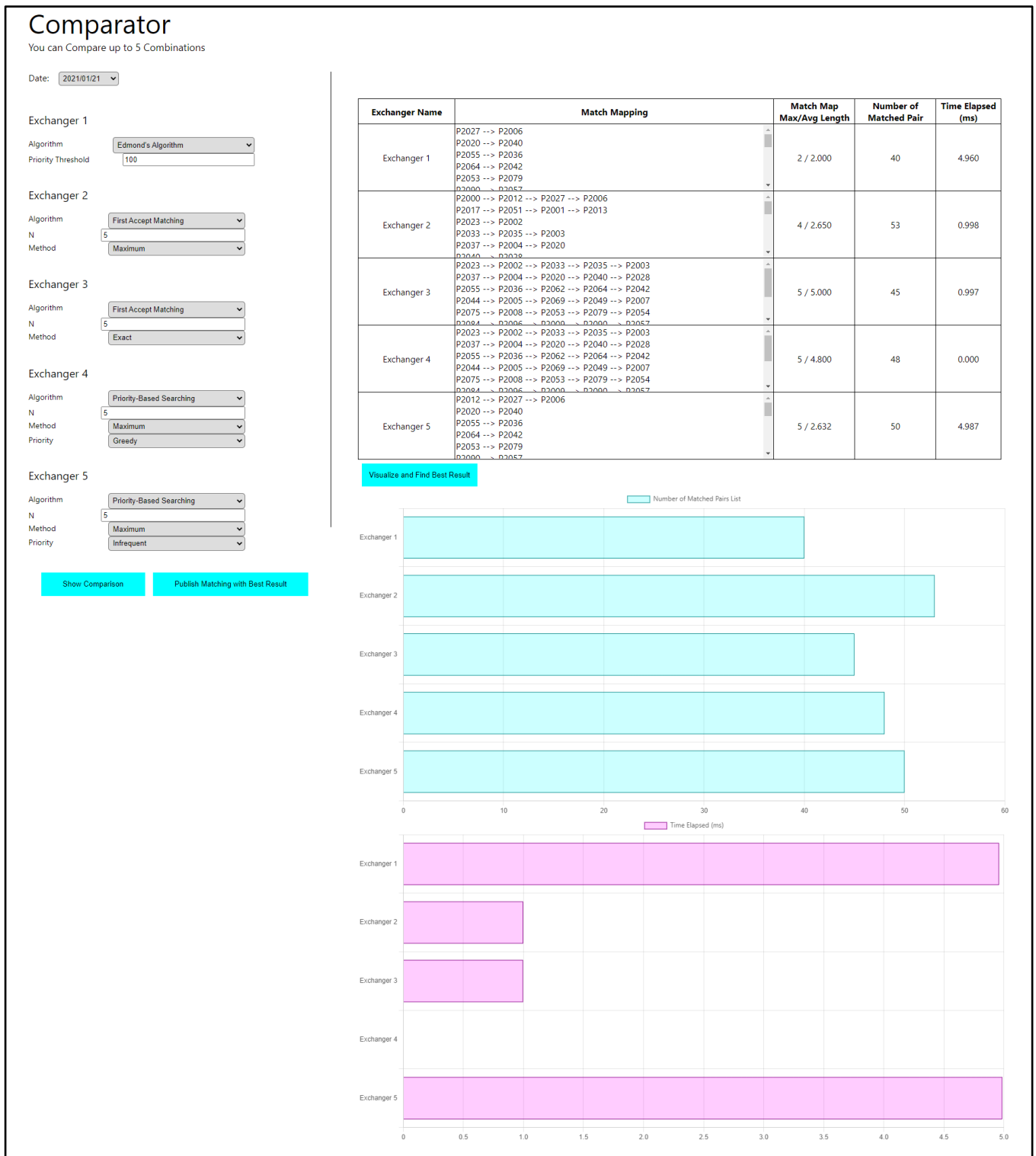
Backend dari aplikasi pengujian memanggil modul-modul hasil implementasi algoritma dan prasyarat-prasyarat yang telah didefinisikan pada subbab III.3. *Library-library* Python yang digunakan untuk implementasi keseluruhan *backend* ini adalah Django 3.2, django-cors-headers 3.7.0, djangorestframework 3.12.4, environs 9.3.2, dan networkx 2.5.1.

IV.1.4.3 Frontend

Antarmuka yang dibuat adalah sebuah *dashboard* sederhana yang akan menampilkan grafik-grafik perbandingan antar kombinasi algoritma dan juga parameter-parameternya. Pengguna dapat membandingkan lima buah kombinasi pada saat yang bersamaan. Untuk mempermudah *input* contoh data medis pasien baru, dibuat suatu *html form* yang dapat secara langsung memasukkan data ke basis data. Implementasi dilakukan menggunakan HTML, CSS, dan juga Vanilla JavaScript karena bersifat ringan dan mudah diimplementasikan. Contoh tampilan *dashboard* saat dilakukan perbandingan dapat dilihat pada Gambar IV.1.3.

IV.1.4.4 Batasan pada Aplikasi Pengujian

Terdapat suatu batasan pada *dashboard* pada implementasi aplikasi yang digunakan untuk pengujian. Dalam suatu waktu, hanya dapat dilakukan perbandingan untuk lima jenis kombinasi yang berbeda. Hal ini dilakukan untuk mencegah *overload* dan mengurangi kompleksitas dari hasil yang perlu dilihat sehingga pengguna tidak perlu melihat terlalu banyak hasil pada layar yang membuat hasil tidak terlihat jelas.



Gambar IV.1.3 Tampilan *Dashboard* saat Membandingkan Lima Kombinasi

IV.2 Pengujian

Tujuan dari pengujian-pengujian yang dilakukan adalah untuk menemukan algoritma dengan parameter-parameter yang mana yang mampu menghasilkan performa terbaik dari metrik-metrik pengujian yang telah didefinisikan. Selain itu, melalui pengujian-pengujian ini dapat dilihat apakah algoritma-algoritma pencarian pemetaan kecocokan yang bersifat N arah mampu menghasilkan performa yang lebih baik dibandingkan dengan algoritma pencarian pemetaan kecocokan dua arah yang digunakan sebagai algoritma *baseline*. Pengujian yang dilakukan dibagi menjadi dua jenis prosedur, prosedur pengujian menggunakan aplikasi dan juga prosedur pengujian menggunakan Jupyter Notebook. Masing-masing prosedur pengujian memiliki kelebihan dan kekurangannya masing-masing.

IV.2.1 Pengujian menggunakan Aplikasi Pengujian

Seperti yang telah dijelaskan pada subbab IV.1.4, diimplementasikan aplikasi pengujian yang digunakan sepenuhnya untuk pengujian. Pengujian-pengujian yang dilakukan pada aplikasi pengujian ini dapat dilakukan dengan cepat karena pengguna aplikasi tidak perlu melakukan pemrograman secara manual. Meskipun lebih cepat, visualisasi-visualisasi dan template hasil perbandingan yang ditampilkan tidak dapat diganti sehingga jika dibutuhkan pengujian yang bersifat lebih fleksibel, tetap digunakan Jupyter Notebook.

IV.2.2 Pengujian menggunakan Jupyter Notebook

Pengujian menggunakan jupyter notebook bersifat merinci namun lebih sulit dikarenakan perlu dilakukan pemrograman manual dan juga diperlukan *library-library* untuk dapat menjalankan beberapa fungsionalitas seperti visualisasi. Pengujian ini dilakukan jika dibutuhkan eksplorasi secara lebih detil dan diperlukannya penggunaan metode-metode visualisasi yang kompleks. Pada perangkat yang digunakan, Python yang digunakan adalah Python 3.8.5 dan Jupyter 1.0.0, dan Notebook 6.3.0.

IV.2.3 Lingkungan Pengujian (*Testing Environment*)

Untuk melakukan testing dan juga membandingkan kinerja algoritma-algoritma pencarian pemetaan kecocokan N arah, *Edmond's Algorithm* juga diimplementasikan sebagai suatu algoritma pencarian pemetaan kecocokan. Untuk menjamin tidak adanya *bias* dari teknologi yang digunakan, implementasi dilakukan dengan bahasa pemrograman Python 3.8.5 untuk semua prasyarat dan juga untuk setiap algoritma pencarian pemetaan kecocokannya. Selain algoritma-algoritmanya, data yang digunakan untuk eksperimen dimasukkan ke suatu basis data yang digunakan oleh semua algoritma. Berikut adalah spesifikasi *hardware* yang digunakan pada lingkungan pengujian ini:

1. OS : Windows 10 Home v1909; 64-bit
2. RAM : 16 GB
3. CPU : Intel Core i7 8750H @ 2.20GHz (12 CPUs)

IV.3 Eksperimen Kombinasi Algoritma dan Parameter

Pada subbab ini dijelaskan secara menyeluruh mengenai eksperimen-eksperimen yang dilakukan untuk mengetahui apakah algoritma pencarian pemetaan kecocokan bersifat N arah dapat mengungguli algoritma pencarian pemetaan kecocokan dua arah yang digunakan sebagai algoritma *baseline*. Untuk mengetahui hal tersebut, perlu diketahui apakah algoritma pencarian pemetaan kecocokan N arah mampu membuahkan hasil dengan waktu eksekusi yang dapat menyaingi algoritma *baseline*. Selain itu, perlu diketahui juga mengenai bagaimana perkembangan waktu eksekusi algoritma pencarian pemetaan kecocokan N arah saat jumlah data yang digunakan semakin bertambah. Selain mengenai waktu eksekusi, tentunya juga dilakukan eksperimen mengenai parameter-parameter terbaik mana yang digunakan pada algoritma pencarian pemetaan kecocokan N arah agar algoritma mampu menghasilkan hasil yang optimal. Pada eksperimen terakhir, dibandingkan *matching efficiency* dari berbagai jenis kombinasi algoritma dan parameter dibandingkan dengan *baseline* untuk melihat kombinasi seperti apa yang mampu menghasilkan *matching efficiency* yang besar tanpa panjang pertukaran yang besar.

Pada eksperimen-eksperimen ini, data yang berisikan 3000 pasangan donor resipien tidak cocok umumnya dipecah menjadi beberapa buah bagian berukuran sama. Setiap bagian tersebut dikatakan sebagai suatu *seed*. Untuk setiap eksperimen dengan pengecualian IV.3.2, data dipisah menjadi 30 *seed* data yang masing-masing berisikan 100 pasangan yang tidak cocok. Eksperimen ini dilakukan secara merinci menggunakan Jupyter Notebook.

IV.3.1 Perbandingan Waktu Eksekusi Setiap Algoritma

Evaluasi waktu eksekusi algoritma dilakukan selama 50 kali untuk beberapa kombinasi algoritma dan parameter yang telah diuji coba dan waktu dicatat setiap kali algoritma pencarian pemetaan kecocokan dijalankan untuk list siklus yang didapatkan dari graf kompatibilitas. Lima puluh kali eksekusi ini dilakukan untuk 30 *seed* data yang telah dijelaskan pada subbab IV.3. Waktu yang dicatat kemudian dirata-ratakan untuk setiap kombinasi algoritma dan parameter yang diuji.

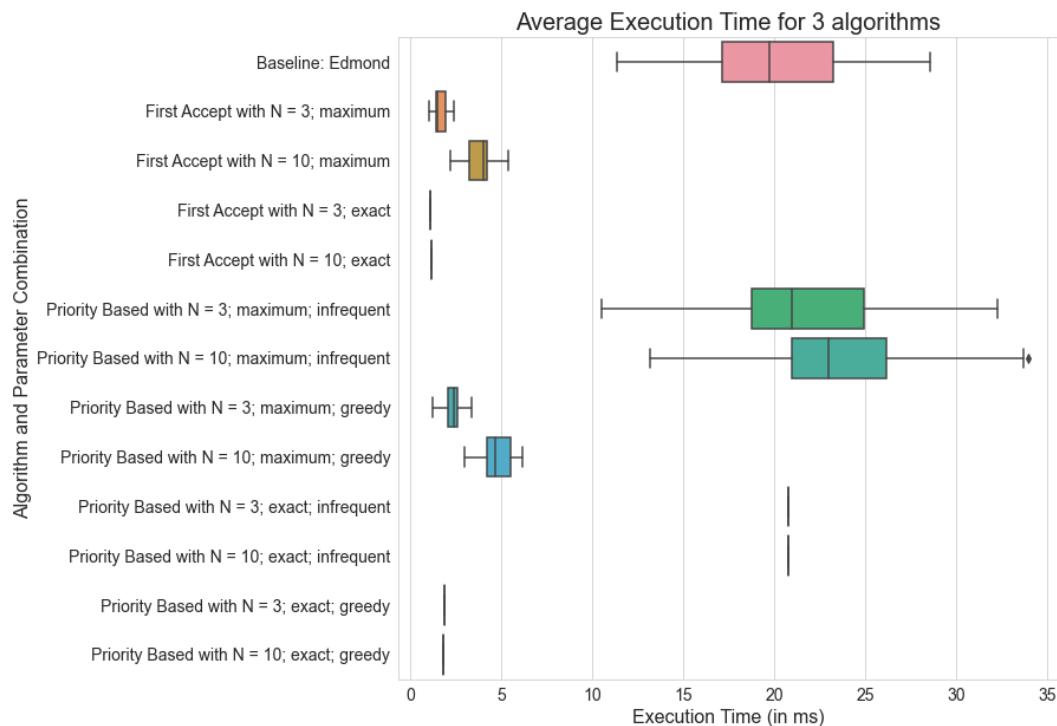
Boxplot yang menyatakan waktu eksekusi rata-rata untuk beberapa kombinasi algoritma dan parameter dalam satuan waktu milisekon (ms) dapat dilihat pada Gambar IV.3.1. Poin-poin data yang ada pada *boxplot* adalah rata-rata waktu eksekusi algoritma pencarian pemetaan kecocokan pada satu *seed* data.

Seperti yang dapat dilihat pada Gambar IV.3.1, waktu eksekusi untuk data yang berbeda tidak menyimpang terlalu jauh satu dengan yang lainnya. Dapat dilihat juga saat algoritma N arah menggunakan metode penggunaan nilai N sebagai panjang eksak, waktu eksekusi dari algoritma tersebut tidak berbeda meskipun data *seed* yang digunakan berbeda.

Meskipun berbeda satu dengan yang lainnya, nilai-nilai dari rata-rata waktu eksekusi algoritma yang dapat terlihat pada grafik tidak memiliki perbedaan yang terhitung sangat lama apabila suatu kombinasi algoritma dan parameter dibandingkan dengan kombinasi lainnya. Sehingga perbandingan waktu antar algoritma dapat dianggap kurang signifikan karena dapat dideduksi secara kasar bahwa algoritma-algoritma pencarian pemetaan kecocokan ini dapat membuahkan hasil pada saat yang kurang lebih bersamaan.

IV.3.2 Perbandingan Waktu Eksekusi untuk Jumlah Data yang Berbeda

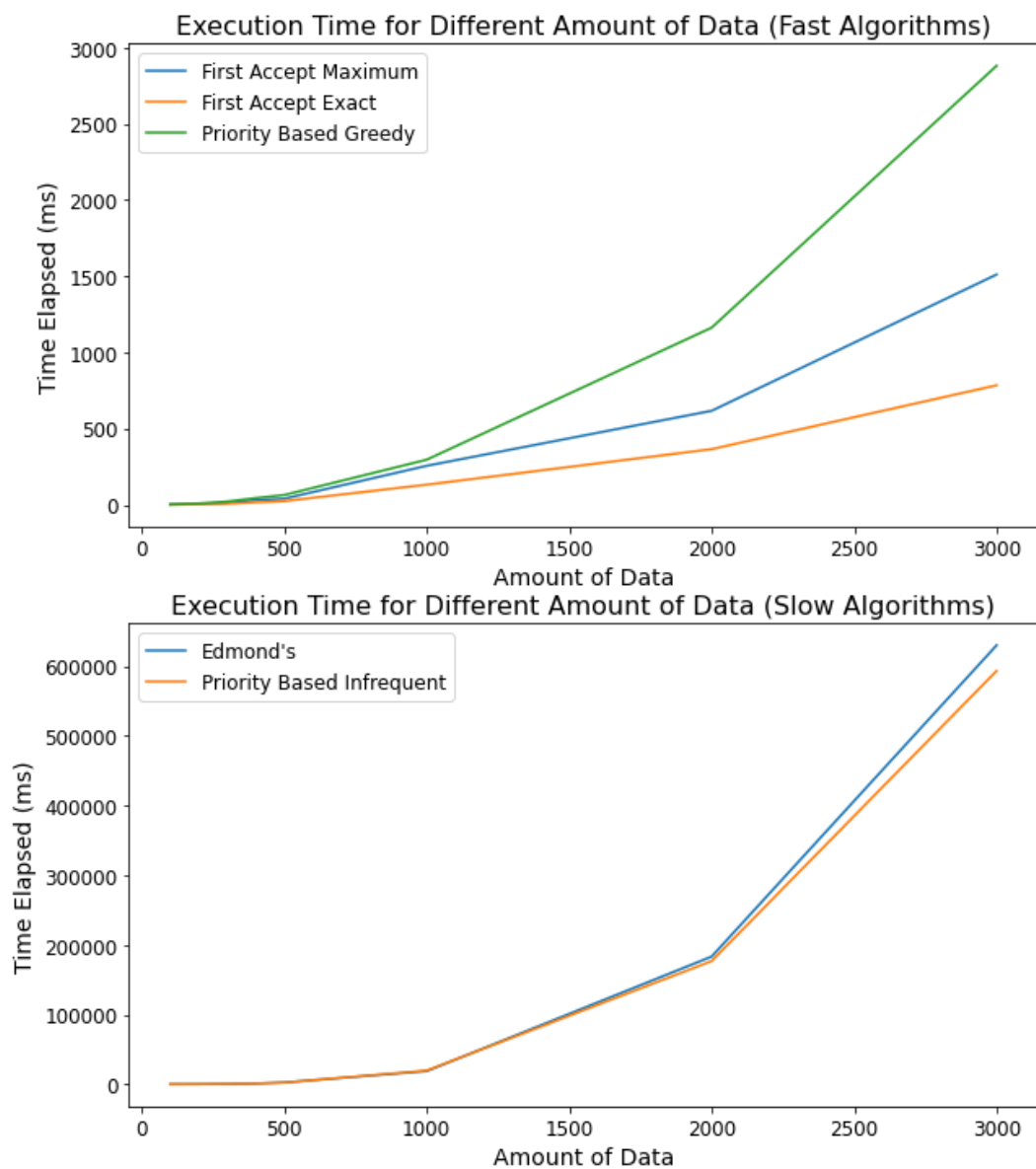
Pengujian berikutnya adalah pengujian untuk mengetahui pengaruh dari jumlah data terhadap waktu eksekusi. Pada tes ini, digunakan *Edmond's algorithm*, *First-accept Search* dengan metode penggunaan nilai N sebagai panjang maksimum dan eksak, *Priority based Search* dengan metode penggunaan nilai N sebagai panjang maksimum dan eksak, dan metode pemilihan prioritas *greedy* dan *infrequent*.



Gambar IV.3.1 Perbandingan Waktu Eksekusi Rata-rata Beberapa Kombinasi

Untuk eksperimen ini, waktu eksekusi yang dicatat adalah rata-rata waktu eksekusi untuk jalannya algoritma menggunakan data sebesar 100, 200, 300, 400, 500, 1000, 2000, dan juga 3000 pasangan. Waktu dicatat setiap kali algoritma pencarian pemetaan kecocokan dijalankan untuk list siklus yang didapatkan dari graf kompatibilitas. Waktu eksekusi rata-rata didapatkan dari algoritma yang dijalankan sebanyak 10 kali untuk setiap jumlah data yang berbeda-beda.

Dapat dilihat pada Gambar IV.3.2, diagram dibagi menjadi dua buah karena adanya interval yang sangat berbeda antara waktu eksekusi algoritma *First Accept Search* dan *Priority-based greedy* terhadap *Edmond's algorithm* dan *Priority-based infrequent*. Kenaikan waktu eksekusi berdasarkan jumlah data bersifat seperti fungsi kuadratik. Hal ini disebabkan karena penggunaan algoritma-algoritma yang memiliki kompleksitas $O(n^2)$ untuk melakukan reduksi list siklus menjadi pemetaan kecocokan.



Gambar IV.3.2 Perbandingan Waktu Eksekusi Rata-rata untuk Jumlah Data yang Berbeda

IV.3.3 Perbandingan Efisiensi Pencocokan berdasarkan Parameter Algoritma yang Digunakan

Dari algoritma-algoritma pencarian pemetaan kecocokan N arah yang telah diimplementasikan, terdapat tiga buah parameter yang dapat diubah dalam jalannya algoritma, yang pertama adalah besar kecilnya nilai N itu sendiri, lalu metode penggunaan nilai N , dan juga metode penentuan prioritas yang merupakan parameter terkhusus untuk algoritma *Priority-based Search*.

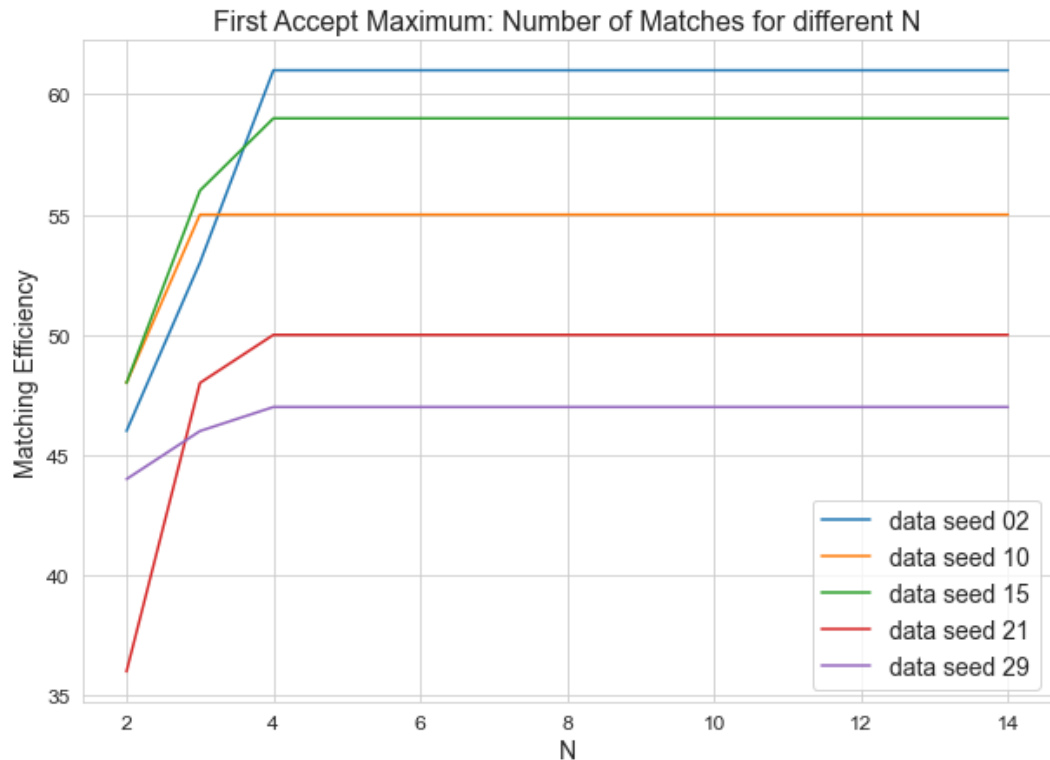
Evaluasi *parameter tuning* ini penting dilakukan karena parameter-parameter yang digunakan pada algoritma mengatur cara jalannya suatu algoritma. Tujuan akhir dari *parameter tuning* ini adalah untuk memperoleh parameter-parameter yang tepat untuk membangun suatu kombinasi parameter untuk algoritma pencarian pemetaan kecocokan yang mampu mendapatkan *matching efficiency* yang besar.

Pada evaluasi ini, dibandingkan parameter-parameter yang digunakan saat algoritma dijalankan menggunakan data yang sama agar dapat diobservasi nilai parameter mana yang lebih unggul untuk mendapatkan *matching efficiency* yang lebih tinggi. Hasil yang diperoleh dari eksperimen-eksperimen pada evaluasi ini adalah nilai *matching efficiency* yang didapatkan saat digunakan algoritma dengan nilai-nilai parameter yang berbeda terhadap data yang sama.

IV.3.3.1 Perbandingan Efisiensi Pencocokan berdasarkan Parameter Nilai N

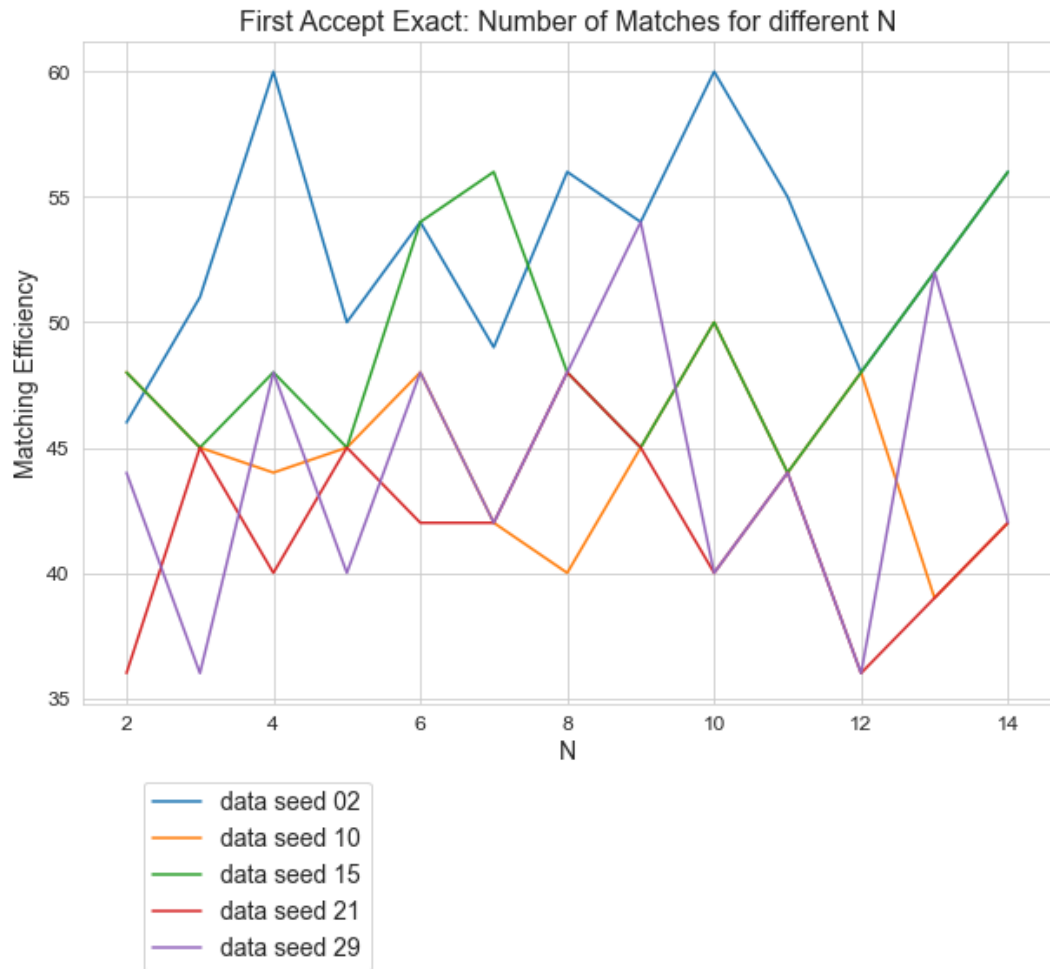
Evaluasi ini dilakukan dengan membandingkan pengaruh nilai N terhadap banyaknya jumlah pasangan yang mendapatkan kecocokan dengan pasangan-pasangan lainnya. Eksperimen dilakukan dengan membandingkan hasil pemetaan kecocokan dari lima buah algoritma masing-masing dengan set yang berisi 100 pasangan. Algoritma dipakai untuk nilai N sebesar 2 hingga 14. Eksperimen dilakukan untuk algoritma *First-Accept Search* dengan metode penggunaan nilai N yang berbeda dan juga algoritma *Priority-based Search* dengan metode penentuan prioritas yang berbeda. Hasil eksperimen dari Algoritma *First-Accept Search* dengan metode *maximum* dapat dilihat pada Gambar IV.3.3.

Seperti yang tercantum pada Gambar IV.3.3, dapat dilihat bahwa semakin besar nilai N , semakin besar *matching efficiency*. Namun tren ini berhenti saat nilai N adalah 5, nilai N yang lebih dari 5 tidak meningkatkan *matching efficiency* yang terjadi. Hal ini mungkin terjadi karena *seed-seed* data yang dipakai dan juga karena limitasi nilai maksimum N yang ada pada eksperimen ini.



Gambar IV.3.3 *Matching Efficiency* untuk Nilai N yang berbeda dengan Algoritma *First-Accept Maximum*

Hasil eksperimen dari Algoritma *First-Accept Search* dengan metode penggunaan nilai N secara eksak dapat dilihat pada Gambar IV.3.4. Tren yang terjadi pada eksperimen yang menggunakan metode maksimum tidak terjadi pada eksperimen yang menggunakan metode eksak. Dapat dilihat pada Gambar IV.3.3 bahwa tidak ada tren apapun yang terjadi. Garis-garis pada grafik bersifat *random* dan tidak ada *insight* pasti yang dapat diambil dari grafik yang ada pada Gambar IV.3.4.

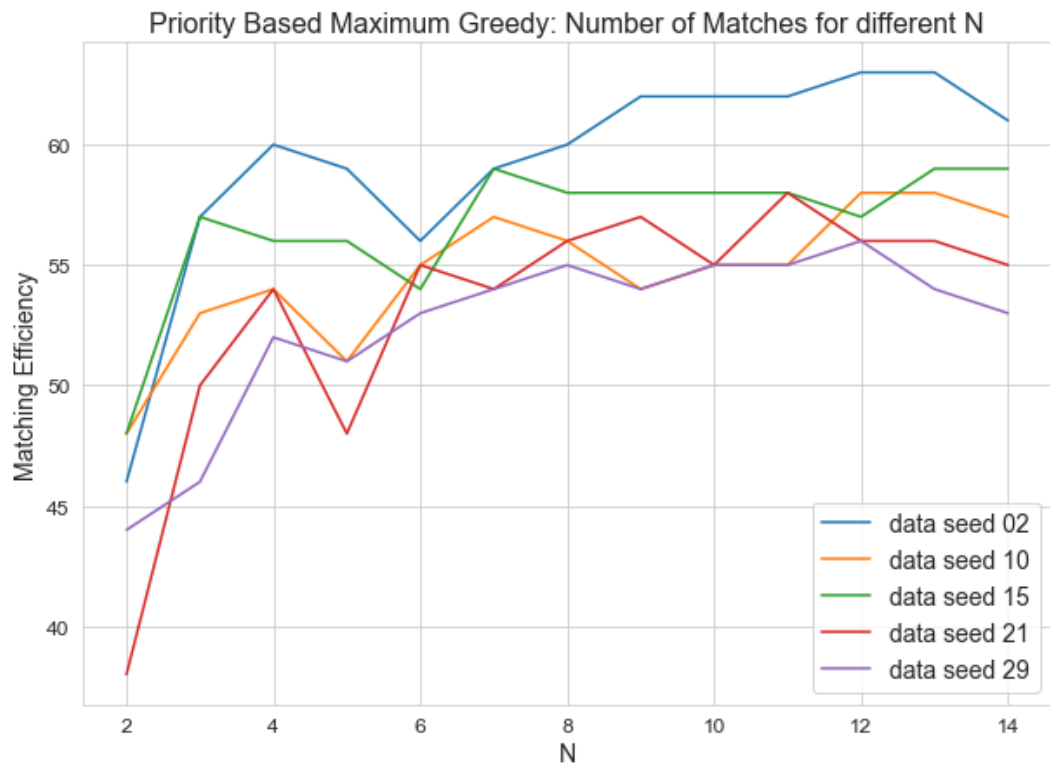


Gambar IV.3.4 *Matching Efficiency* untuk Nilai N yang berbeda dengan Algoritma *First-Accept Exact*

Hasil eksperimen dari Algoritma *Priority-based Search* dengan metode penggunaan nilai N sebagai panjang maksimum dan penentuan prioritas *greedy* dapat dilihat pada Gambar IV.3.5. Seperti yang dapat dilihat pada gambar IV.3.5, *matching efficiency* yang didapat relatif meningkat seiring membesarnya nilai N . Banyak naik-turun yang terjadi di tengah tren, namun secara umum, tren bersifat menaik.

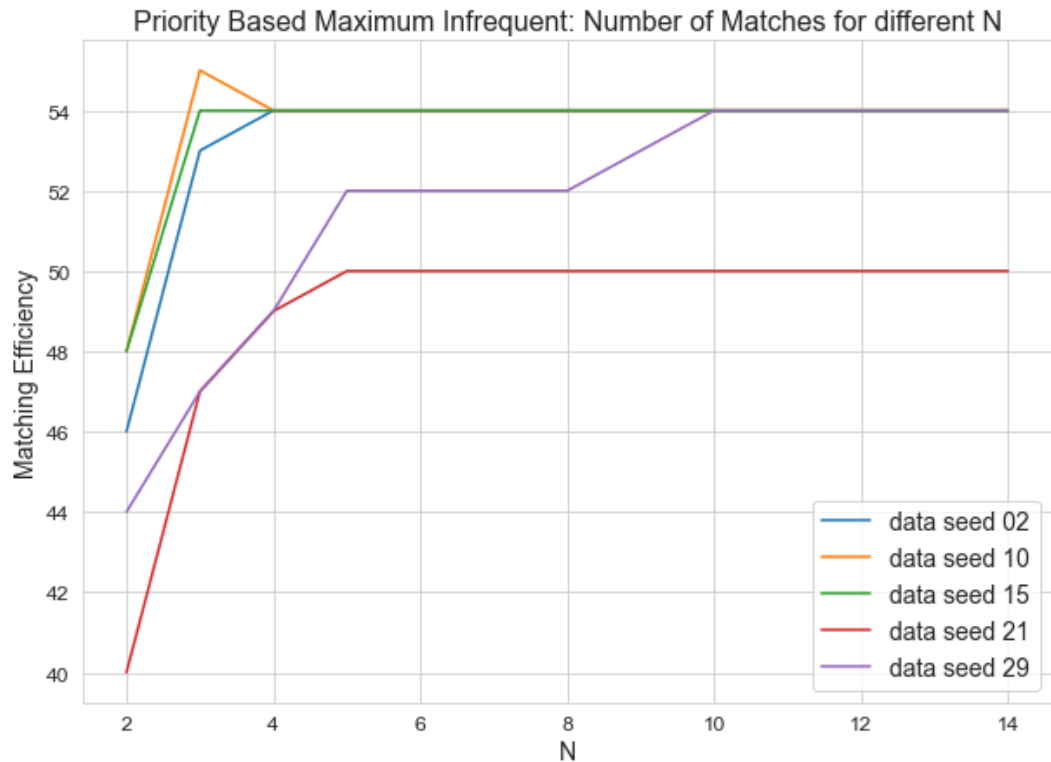
Hasil eksperimen dari Algoritma *Priority-based Search* dengan metode penggunaan nilai N sebagai panjang maksimum dan penentuan prioritas *infrequent* dapat dilihat pada Gambar IV.3.6. Tidak seperti eksperimen yang menggunakan

metode penentuan prioritas *greedy*, setelah nilai N menyentuh angka 10, keempat kasus ini tidak meningkatkan *matching efficiency*. Tren yang bersifat menaik masih berlaku meskipun terdapat satu buah penurunan saat menggunakan data seed 10 dari nilai N yang adalah 4 ke 5.



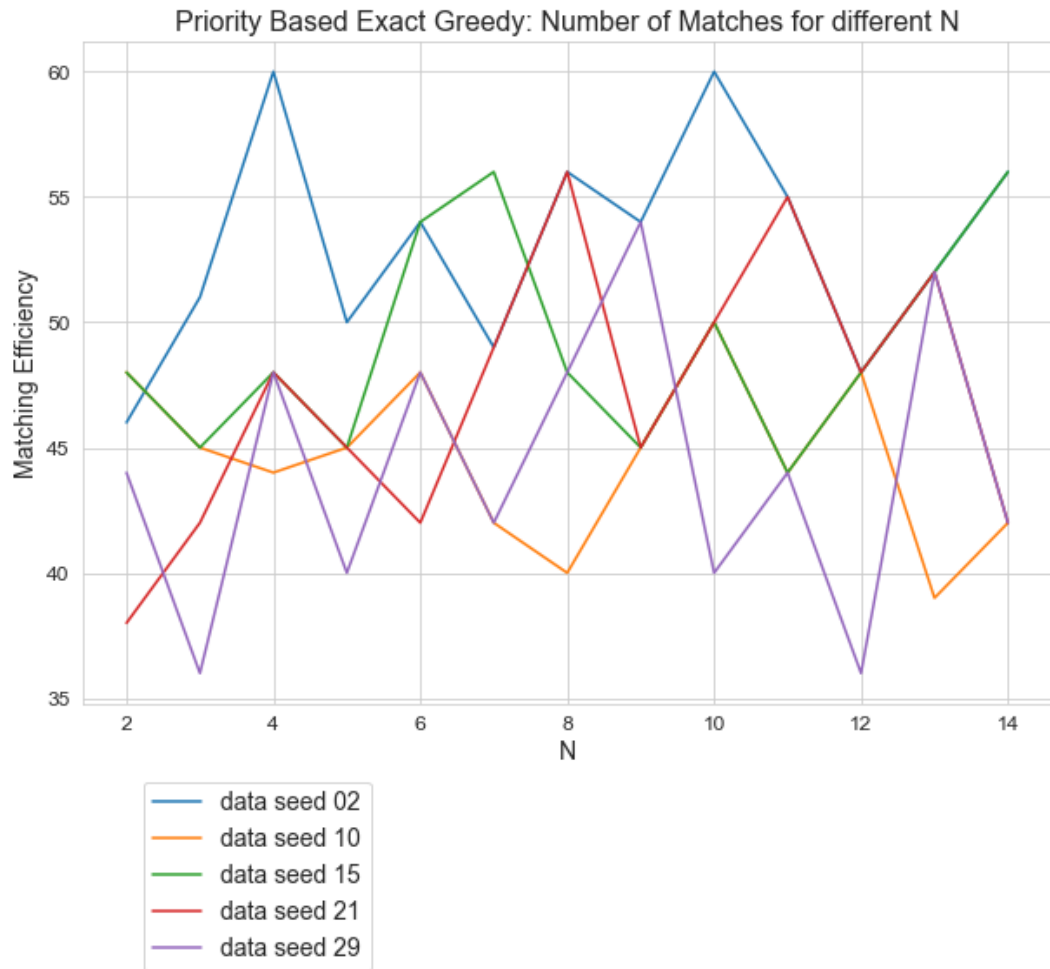
Gambar IV.3.5 *Matching Efficiency* untuk Nilai N yang berbeda dengan Algoritma *Priority-based Maximum Greedy*

Hasil eksperimen dari Algoritma *Priority-based Search* dengan metode penggunaan nilai N sebagai panjang eksak dan penentuan prioritas *greedy* dapat dilihat pada Gambar IV.3.7. Seperti yang dapat dilihat pada gambar IV.3.7, *matching efficiency* yang terjadi sangat *random* untuk nilai N yang berbeda, tidak ada tren yang terjadi, seperti saat digunakan metode penggunaan nilai N sebagai panjang eksak pada algoritma *First-accept Searching*.



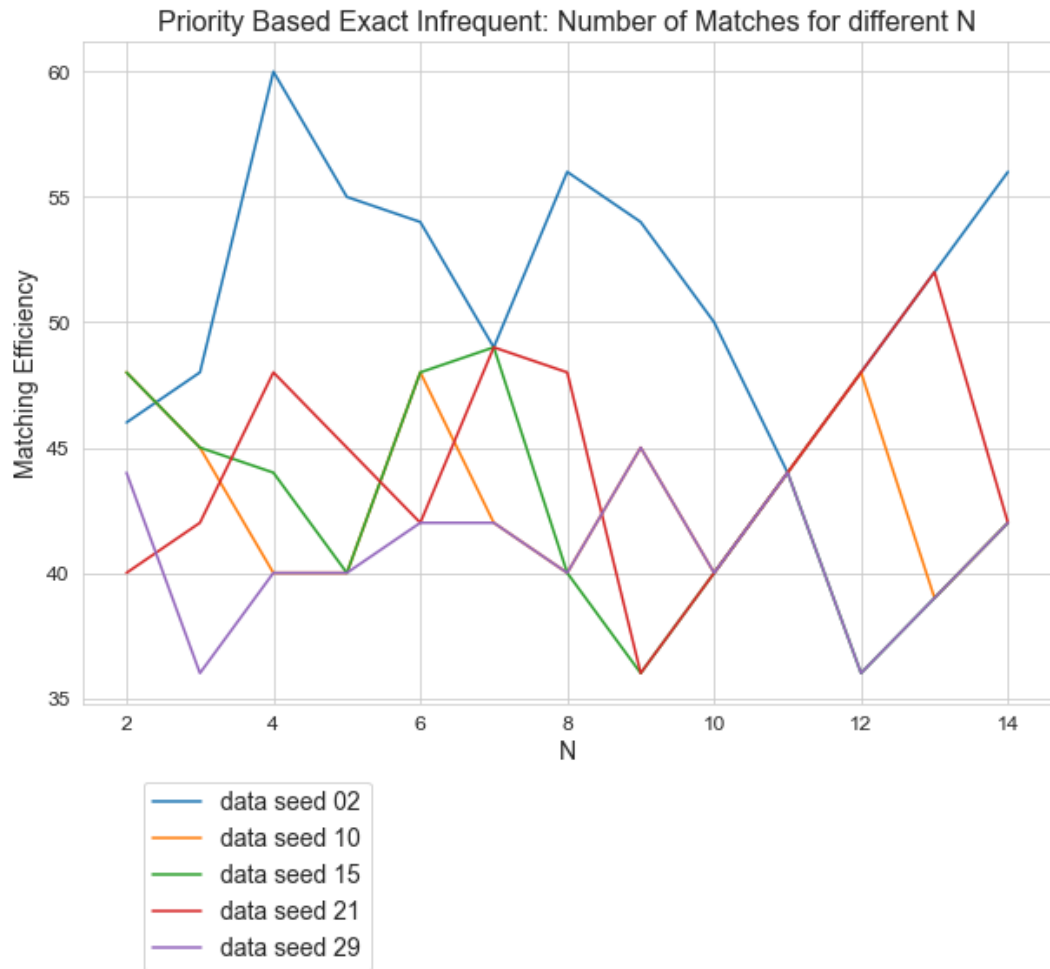
Gambar IV.3.6 *Matching Efficiency* untuk Nilai N yang berbeda dengan Algoritma *Priority-based Maximum Infrequent*

Hasil eksperimen dari Algoritma *Priority-based Search* dengan metode penggunaan nilai N sebagai panjang eksak dan penentuan prioritas *infrequent* dapat dilihat pada Gambar IV.3.8. Seperti yang dapat dilihat pada Gambar IV.3.8, *matching efficiency* yang terjadi sangat *random* untuk nilai N yang berbeda, tidak ada tren yang terjadi, seperti saat digunakan metode penggunaan nilai N sebagai panjang eksak pada algoritma *First-accept Searching* dan *Priority-based Search* dengan metode penggunaan nilai N sebagai panjang eksak dan penentuan prioritas *greedy*.



Gambar IV.3.7 *Matching Efficiency* untuk Nilai N yang berbeda dengan Algoritma *Priority-based Exact Greedy*

Berdasarkan hasil setiap eksperimen, secara umum didapatkan deduksi bahwa semakin besar nilai N , maka kemungkinan besar semakin besar juga *matching efficiency* yang dapat diperoleh, niscaya lebih banyak pasien yang dapat terselamatkan. Meskipun kekuatan operasional setiap rumah sakit berbeda dari satu dengan yang lainnya, meningkatkan jumlah pertukaran dapat mengartikan lebih banyaknya jumlah pasien berkebutuhan ginjal cangkok yang dapat mendapatkan transplantasi.



Gambar IV.3.8 *Matching Efficiency* untuk Nilai N yang berbeda dengan Algoritma *Priority-based Exact Infrequent*

IV.3.3.2 Perbandingan Efisiensi Pencocokan berdasarkan Parameter Metode Penggunaan Nilai N

Karena metode penggunaan nilai N juga merupakan suatu parameter pada algoritma *Priority-based Search*, eksperimen dilakukan menggunakan algoritma *First-Accept Search*, *Priority-based Search* dengan prioritas *greedy*, dan *Priority-based Search* dengan prioritas *infrequent*. Seperti yang telah dijelaskan pada subbab III.2, pemilihan nilai N yang pertama adalah untuk menggunakan nilai N sebagai batas maksimum, sehingga pertukaran yang memiliki arah kurang dari N tetap dapat terpilih. Tidak seperti metode maksimum, metode eksak menggunakan nilai N

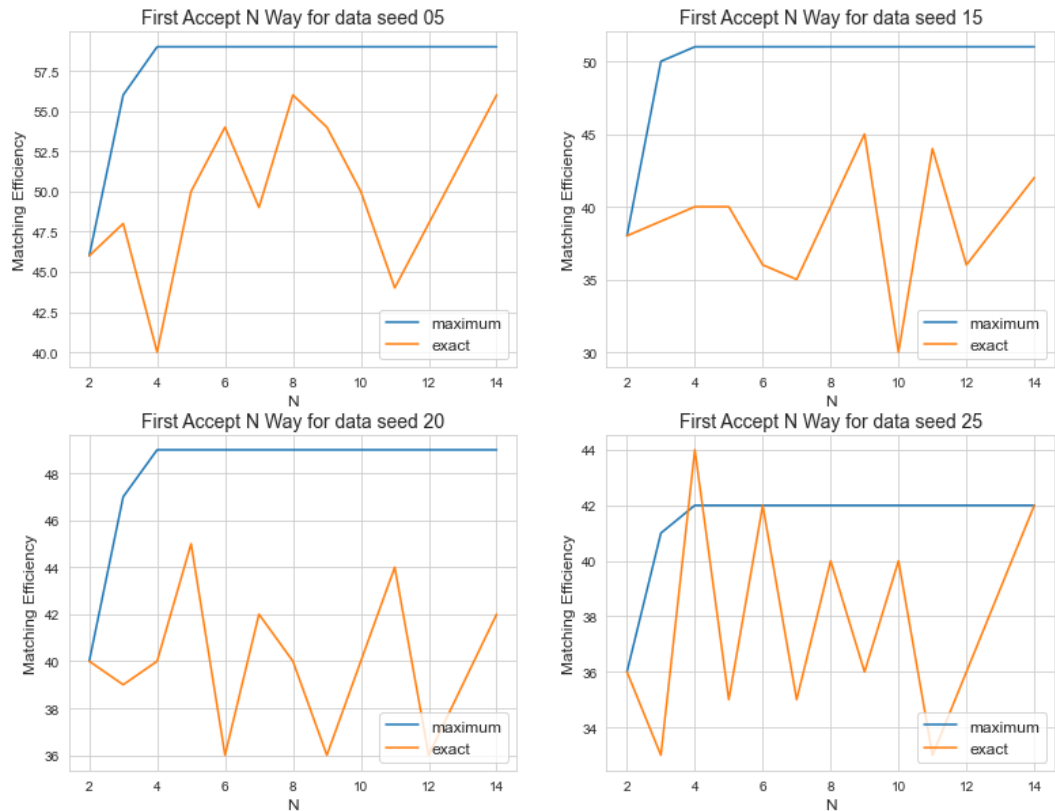
sebagai nilai pasti dalam penggunaan nilai N . Pertukaran yang mungkin terjadi dengan jumlah arah yang tidak sama dengan N tidak akan dilakukan. Hasil eksperimen didapatkan menggunakan empat data *seed* yang berbeda dengan nilai N dari 2 hingga 14 yang secara jelas dapat dilihat pada Gambar IV.3.9, Gambar IV.3.10, dan Gambar IV.3.11.

Pada Gambar IV.3.9, Gambar IV.3.10, dan juga Gambar IV.3.11, tren yang sama terjadi, yaitu metode *maximum* unggul untuk hampir semua nilai N dan konsisten untuk keempat data yang digunakan. Meskipun pada beberapa kasus khusus, penggunaan metode eksak mampu menyaingi metode maksimum seperti pada grafik keempat saat nilai N adalah 4 dimana *matching efficiency* yang didapatkan oleh algoritma yang menggunakan metode penggunaan nilai N sebagai panjang eksak mengungguli *matching efficiency* yang didapatkan oleh algoritma yang metode penggunaan nilai N sebagai panjang maksimum.

IV.3.3.3 Perbandingan Efisiensi Pencocokan berdasarkan Parameter Metode Penentuan Prioritas

Seperti yang telah dijelaskan pada subbab III.2, terdapat dua jenis metode penentuan prioritas pada algoritma *Priority-based Searching*. Penentuan pertama adalah dengan metode *greedy* atau mengambil pertukaran-pertukaran dengan arah yang lebih banyak terlebih dahulu. Penentuan kedua adalah dengan metode *infrequent* atau mengambil pertukaran-pertukaran dengan pasangan-pasangan dengan pasien yang lebih berkebutuhan khusus terlebih dahulu.

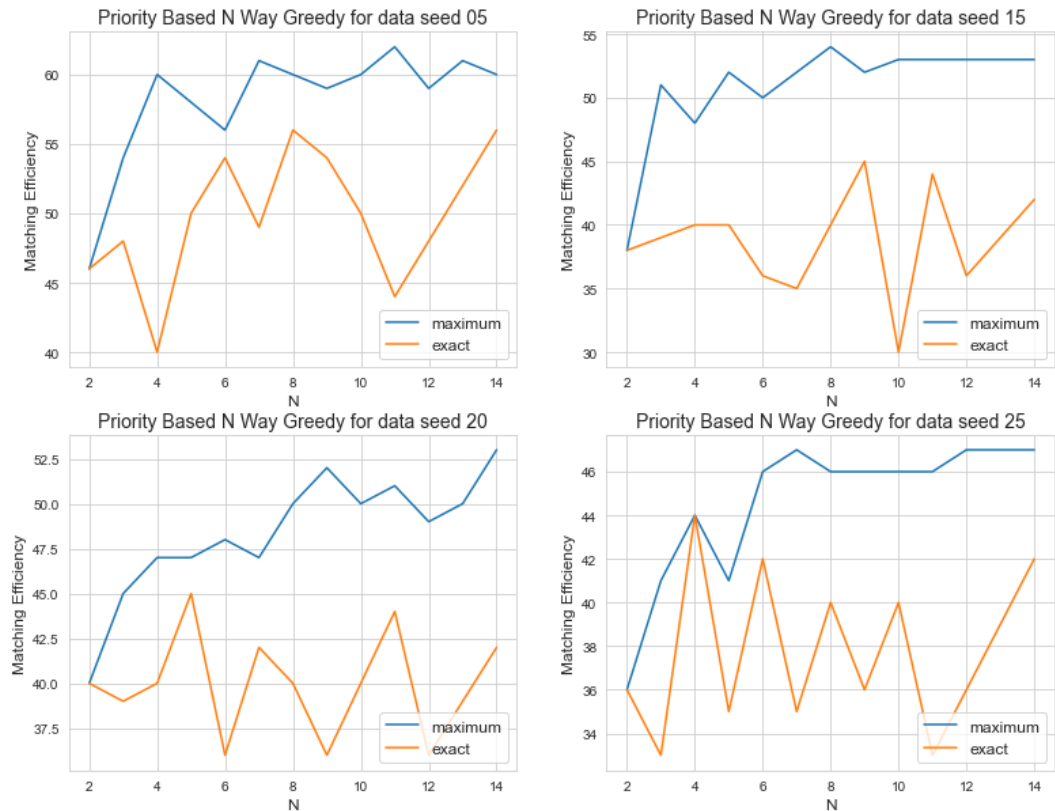
Hasil eksperimen didapatkan menggunakan empat *seed* data yang berbeda dengan nilai N dari 2 hingga 14 yang secara jelas dapat dilihat pada Gambar IV.3.12 dan Gambar IV.3.13 untuk kedua metode penggunaan nilai N . Seperti yang dapat dilihat pada Gambar IV.3.12, secara konsisten penentuan prioritas *greedy* mengungguli penentuan prioritas *infrequent*.



Gambar IV.3.9 Perbandingan Metode Penggunaan Nilai N pada Algoritma *First Accept Search*

Hal ini sesuai dengan ekspektasi awal karena dengan menggunakan metode penentuan prioritas *greedy*, diharapkan jumlah kecocokan yang terjadi lebih banyak sementara dengan menggunakan metode penentuan prioritas *infrequent*, jumlah kecocokan tidak terlalu banyak karena memprioritaskan pasangan-pasangan yang lebih langka terlebih dahulu.

Untuk gambar IV.3.13, hasil yang didapatkan tidak konsisten untuk data yang berbeda. Hal ini disebabkan oleh metode penggunaan nilai N sebagai panjang eksak yang digunakan. Seperti yang dapat dilihat pada eksperimen pada subbab IV.3.4, penggunaan nilai N sebagai panjang eksak tidak memiliki tren.

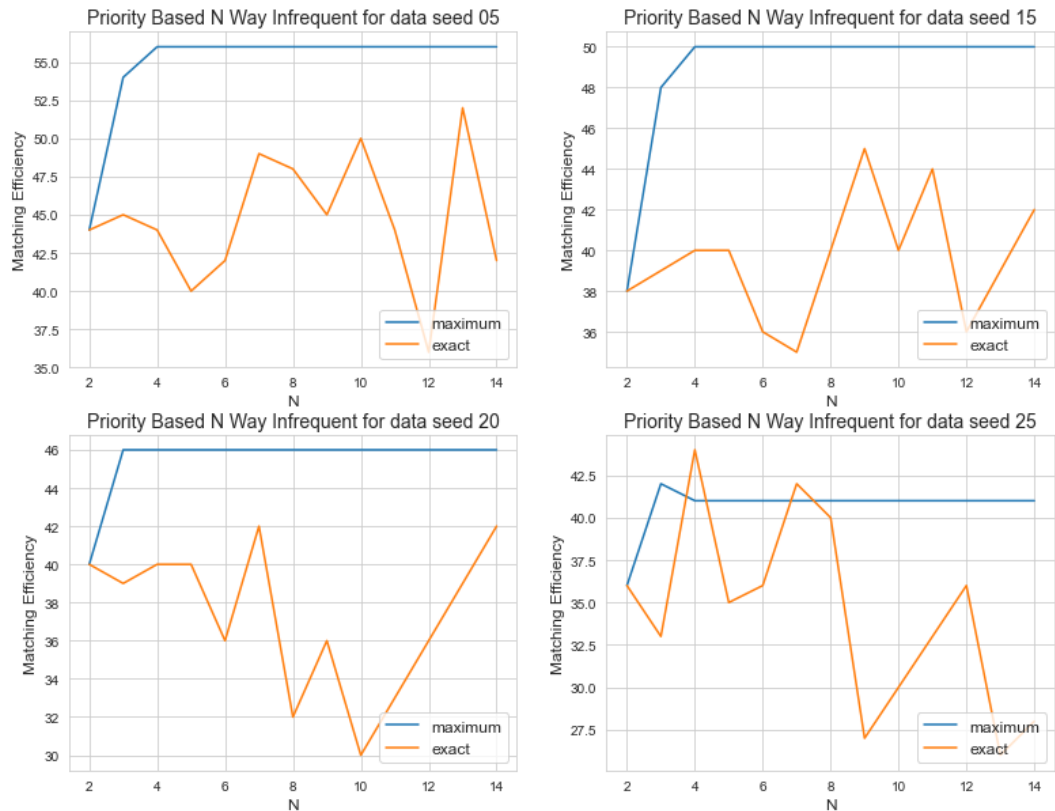


Gambar IV.3.10 Perbandingan Metode Penggunaan Nilai N pada Algoritma *Priority Based Search Greedy*

IV.3.4 Perbandingan Kombinasi Algoritma dan Parameter

Eksperimen terakhir adalah eksperimen untuk melihat kombinasi algoritma dan parameter manakah yang terbaik. Terbaik yang dinilai dari eksperimen ini adalah jumlah kecocokan yang didapatkan oleh kombinasi, arah pencocokan maksimum yang diperoleh kombinasi, dan juga arah pencocokan rata-rata yang diperoleh kombinasi.

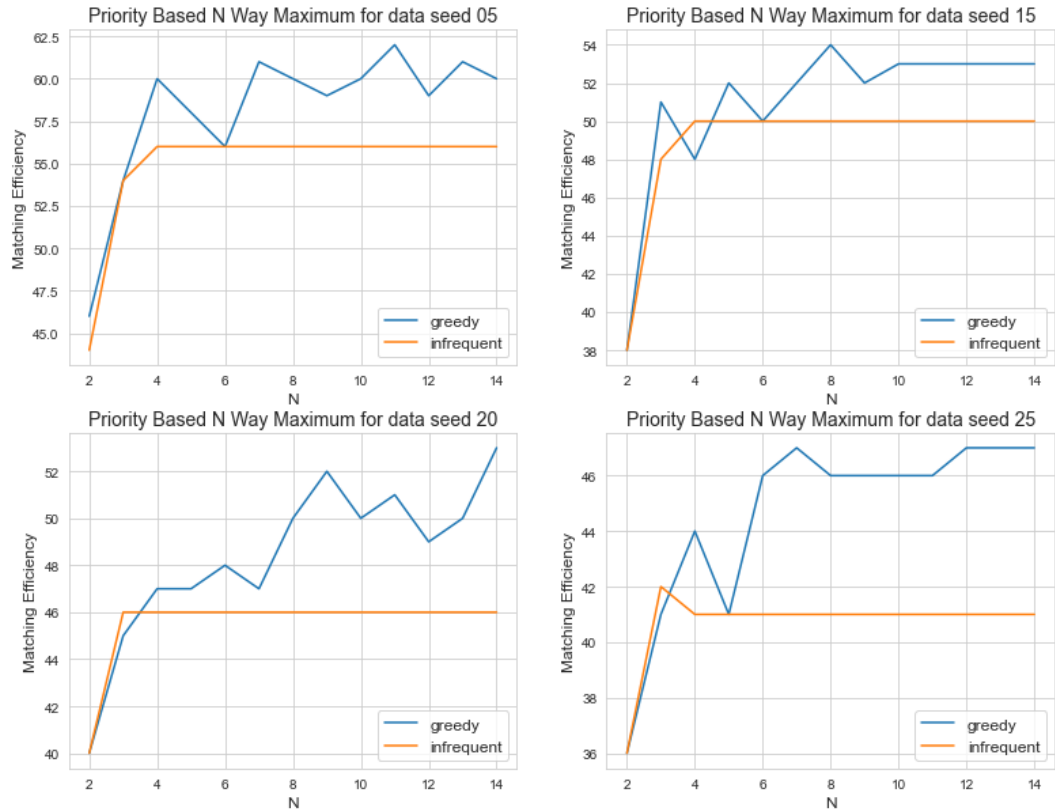
Alasan dari pemilihan pengukuran ini adalah untuk mendapatkan kombinasi algoritma dan parameter yang mampu menghasilkan jumlah maksimum kecocokan dan pada saat yang bersamaan, menggunakan sumber daya operasional rumah sakit minimum. Kombinasi akan dicobakan untuk setiap tiga puluh buah *data seed* yang masing-masing berisikan 100 buah pasangan ketidakcocokan.



Gambar IV.3.11 Perbandingan Metode Penggunaan Nilai N pada Algoritma *Priority Based Search Infrequent*

Hasil dari eksperimen ini disimpan pada suatu tabel dengan kolom untuk setiap kombinasi algoritma dan baris untuk setiap *seed* data. Setiap *cell* data menandakan hasil berupa *matching efficiency* yang diperoleh oleh suatu kombinasi algoritma dan parameter untuk suatu *seed* data. Hasil detil dari eksperimen untuk mencari jumlah kecocokan terbanyak dapat dilihat pada Lampiran B.

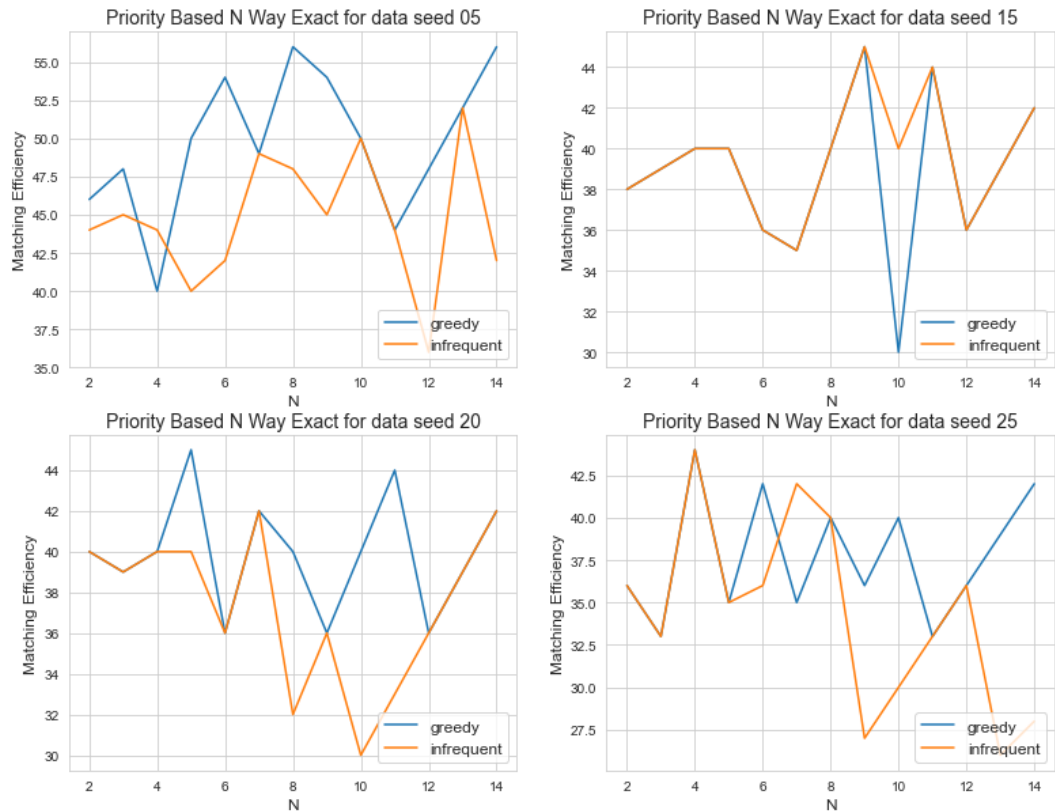
Untuk mempermudah perbandingan antara algoritma-algoritma pencarian pemetaan kecocokan N arah dengan *baseline*, hasil yang tercantum pada Lampiran B diubah untuk menggunakan rasio perbandingan *matching efficiency* pada algoritma *baseline* dan algoritma-algoritma lainnya. Perbandingan secara detil ini dapat dilihat pada Lampiran C. Rata-rata rasio *matching efficiency* untuk setiap kombinasi algoritma terhadap *baseline* dapat dilihat pada Tabel IV.3.1. Visualisasi dari perbandingan ini dapat dilihat pada Gambar IV.3.14.



Gambar IV.3.12 Perbandingan Metode Penentuan Prioritas pada Algoritma *Priority Based Search Maximum*

Seperti yang dapat dilihat pada Tabel IV.3.1, secara umum, dapat dilihat bahwa algoritma pencarian pemetaan kecocokan N arah mampu menghasilkan peningkatan *matching efficiency* untuk mayoritas jenis kombinasi. Meskipun begitu, terdapat empat buah kombinasi yang diujikan yang menghasilkan performa yang lebih buruk dibandingkan algoritma *baseline*.

Rata-rata peningkatan *matching efficiency* saat digunakan algoritma pencarian pemetaan kecocokan N arah dibandingkan .saat digunakan algoritma *baseline* adalah sebesar 7.75%. Dapat dilihat pada Lampiran C, peningkatan *matching efficiency* terbesar terjadi saat algoritma *First-accept Match* digunakan dengan N bernilai 10 dan metode penggunaan nilai N sebagai panjang maksimum, menggunakan data *seed* 21, dimana peningkatan yang terjadi sebesar 47.06%.



Gambar IV.3.13 Perbandingan Metode Penentuan Prioritas pada Algoritma *Priority Based Search Exact*

Hasil dari eksperimen untuk mencari kombinasi yang dapat memperoleh arah pencocokan maksimum terkecil dapat dilihat pada Gambar IV.3.15. Dapat dilihat pada Gambar IV.3.15 bahwa kombinasi yang memiliki arah pencocokan maksimum terkecil adalah algoritma *baseline*, yaitu *Edmond's algorithm*.

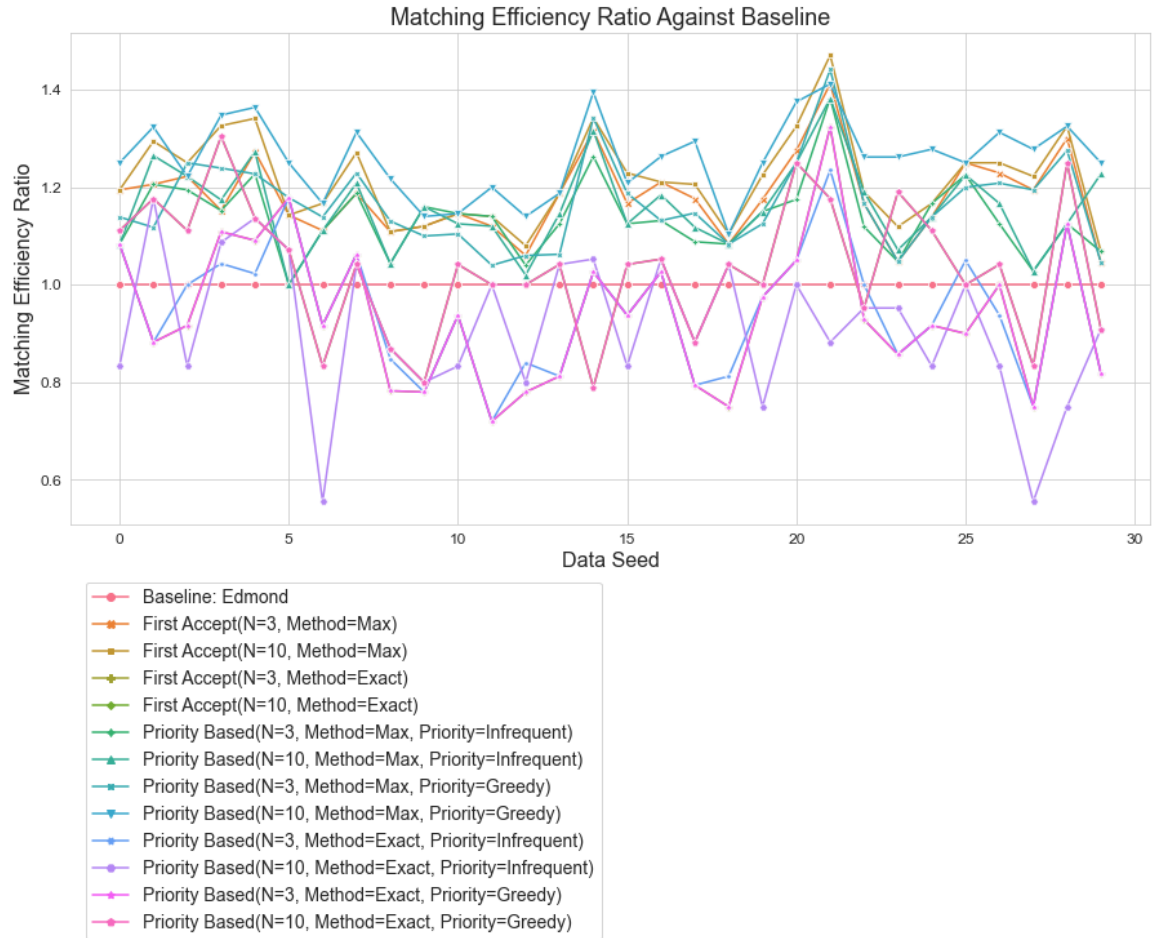
Hal ini sesuai ekspektasi karena *Edmond's algorithm* adalah algoritma pencarian pemetaan kecocokan yang bersifat dua arah, sehingga arah pencocokan yang terjadi sudah pasti adalah dua arah. Untuk kombinasi-kombinasi yang menggunakan algoritma pencarian pemetaan kecocokan N arah, arah pencocokan maksimum terkecil yang adalah 3 didapatkan beberapa kombinasi yang berbeda.

Tabel IV.3.1 Rata-rata Rasio Peningkatan *Matching Efficiency* Algoritma *N* Arah dibandingkan dengan Algoritma *Baseline*

Algoritma	Rata-rata Rasio Perkembangan
Baseline: Edmond	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

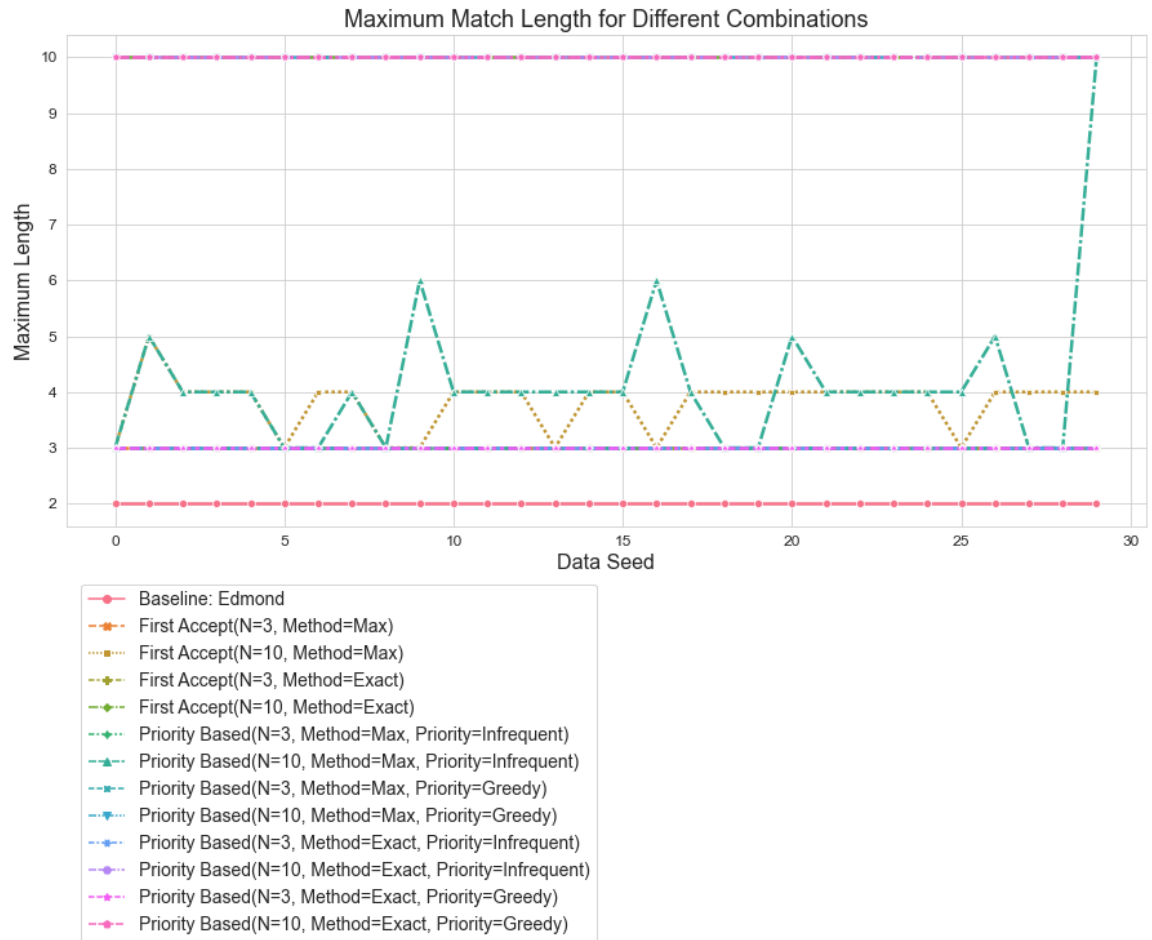
Hasil dari eksperimen untuk mencari kombinasi yang dapat memperoleh arah pencocokan rata-rata terkecil dapat dilihat pada Gambar IV.3.16. Seperti yang dapat dilihat pada Gambar IV.3.16, mayoritas kombinasi memiliki rata-rata arah pencocokan sekitar 2 hingga 3. Dengan pengecualian algoritma *First Accept* dengan nilai *N* 10 dan metode penggunaan nilai *N* sebagai panjang eksak, dan algoritma *Priority-based* dengan *N* 10 dan metode penentuan prioritas *greedy*.

Untuk kombinasi yang menggunakan algoritma *First-accept Searching*, hal ini terjadi karena setiap pencocokan yang diambil dari suatu *list siklus* harus sepanjang *N*, sehingga arah pencocokan rata-rata yang terjadi sudah pasti bernilai *N*, dimana nilai *N* disini adalah 10.



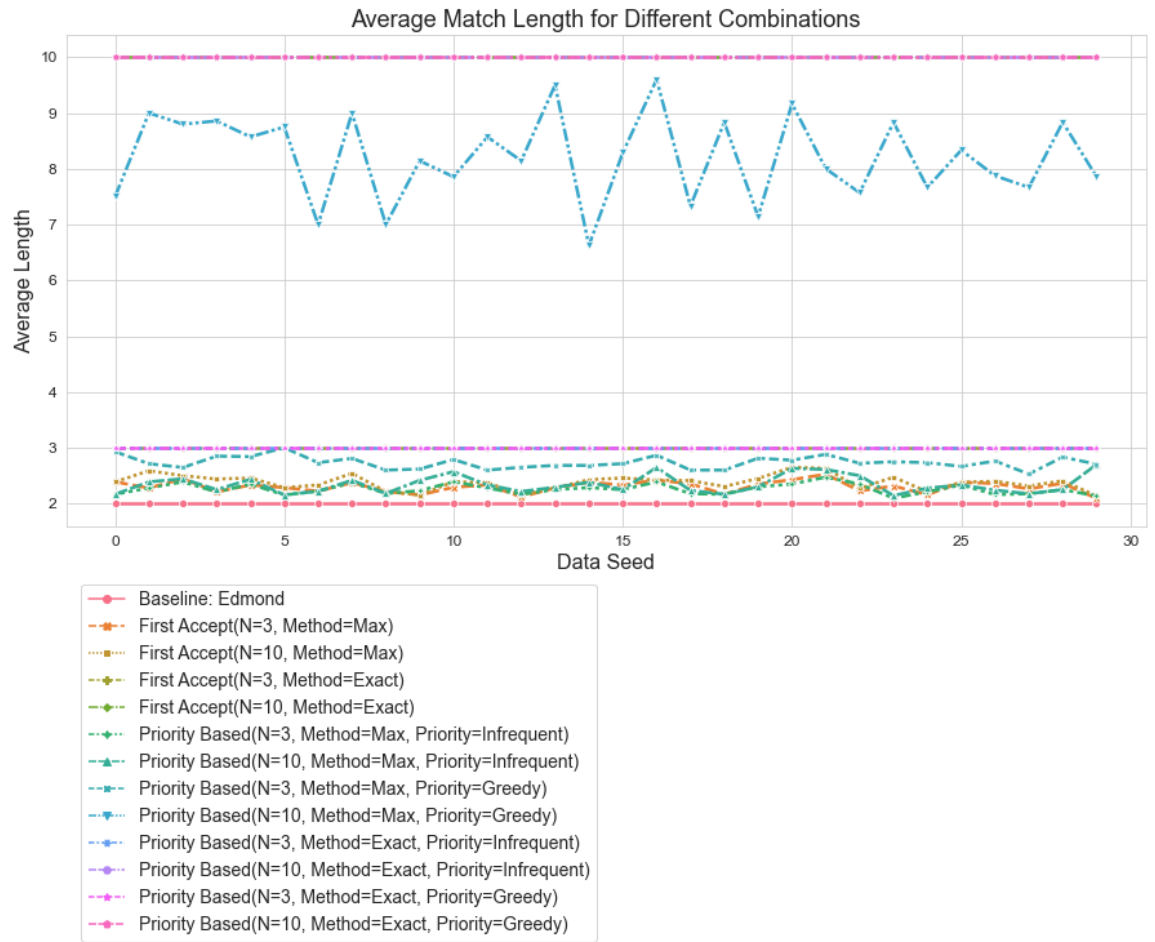
Gambar IV.3.14 Rasio Perbandingan Algoritma-algoritma N arah dengan *Baseline: Edmond*

Untuk kombinasi yang menggunakan algoritma *Priority-based Searching*, hal ini terjadi karena siklus-siklus yang diambil adalah siklus-siklus terpanjang dari list siklus pada graf kompatibilitas, sehingga rata-rata arah pencocokan yang terbentuk relatif tinggi. Berdasarkan tiga visualisasi yang didapatkan pada eksperimen ini, Gambar IV.3.14, Gambar IV.3.15, Gambar IV.3.16, dapat dilihat bahwa kombinasi terbaik yang menghasilkan jumlah kecocokan yang besar dengan arah pencocokan yang cenderung kecil adalah kombinasi menggunakan algoritma *First-accept Searching* dengan nilai N 10 dan metode penggunaan nilai N sebagai panjang maksimum.



Gambar IV.3.15 Arah Pencocokan Maksimum untuk Kombinasi yang Berbeda

Algoritma ini berhasil mendapatkan jumlah kecocokan relatif lebih besar dibandingkan kebanyakan kombinasi lainnya dan juga mampu memperoleh arah pencocokan maksimum dan rata-rata yang relatif lebih kecil dibandingkan kebanyakan kombinasi lain.



Gambar IV.3.16 Arah Pencocokan Rata-rata untuk Kombinasi yang Berbeda

BAB V

KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan-kesimpulan yang dapat ditarik berdasarkan eksperimen dan juga saran-saran untuk pengembangan-pengembangan berikutnya.

V.1 Kesimpulan

Berdasarkan hasil-hasil eksperimen yang diperoleh, didapatkan beberapa kesimpulan dari tugas akhir ini yaitu:

1. Algoritma-algoritma pencarian pemetaan kecocokan N arah yang secara konsisten berhasil memperoleh *matching efficiency* yang lebih tinggi dibandingkan algoritma pencarian pemetaan kecocokan dua arah (algoritma *baseline*) merupakan algoritma *First-accept Searching* dengan metode penggunaan nilai N yang manapun dan juga algoritma *Priority based Searching* jika digunakan metode penggunaan nilai N maksimum, dan metode penentuan prioritas manapun. Peningkatan *matching efficiency* dibandingkan *baseline* secara rata-rata mencapai 7.75% secara umum.
2. Secara umum, hasil dari algoritma *First-accept Searching* baik dengan metode penggunaan nilai N sebagai panjang maksimum maupun eksak dapat memperoleh *matching efficiency* yang lebih tinggi dibandingkan algoritma *baseline*.
3. Secara umum, hasil dari algoritma *Priority based Searching* dengan metode penggunaan nilai N sebagai panjang maksimum dan metode penentuan prioritas baik *greedy* maupun *infrequent* mampu menghasilkan *matching efficiency* yang lebih tinggi dibandingkan algoritma *baseline*. Namun saat metode penggunaan nilai N yang dipakai adalah eksak, *matching efficiency* yang diperoleh lebih rendah dibandingkan algoritma *baseline* dan hanya lebih tinggi pada kasus-kasus tertentu saja.
4. Jika dinilai dari algoritma yang menghasilkan *matching efficiency* tertinggi, algoritma pencarian pemetaan kecocokan N arah terbaik adalah algoritma

Priority based Searching. Meskipun berperforma buruk saat menggunakan metode penggunaan nilai N sebagai panjang eksak, *matching efficiency* yang didapatkan algoritma *Priority-based Searching* saat menggunakan nilai N sebagai panjang maksimum memperoleh nilai-nilai yang sangat tinggi jika dibandingkan dengan algoritma *First-accept Searching*.

5. Jika dinilai dari konsistensi, algoritma *First-accept Searching* unggul dibandingkan algoritma *Priority based Searching* karena algoritma *First-accept Searching* mampu memperoleh *matching efficiency* yang secara konsisten lebih tinggi dibandingkan *baseline* menggunakan parameter yang berbeda-beda.

V.2 Saran

Terdapat suatu saran yang dapat dijadikan sebagai bahan pertimbangan dalam pengembangan lebih lanjut untuk studi-studi sejenis. Untuk pengembangan lebih lanjut, dapat dilakukan eksplorasi mengenai paralelisme untuk algoritma pencarian pemetaan kecocokan agar algoritma dapat berjalan lebih cepat dan dapat menjadi lebih *scalable* untuk data berukuran besar.

DAFTAR REFERENSI

- 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>
- Aprilano, W. D. (2021, January 20). *Teknik Transplantasi Ginjal*. Retrieved from alomedika.com: <https://www.alomedika.com/tindakan-medis/transplantasi/transplantasi-ginjal/teknik>
- Chargé, S., & Hodgkinson, K. (2017, January). *Blood: the basics*. Retrieved from profedu.blood.ca/: <https://profedu.blood.ca/en/transfusion/publications/blood-basics>
- Mehta, D. (2020, May 27). *Detect Cycle in a Directed Graph*. Retrieved from geeksforgeeks.org: <https://www.geeksforgeeks.org/detect-cycle-in-a-graph/>
- 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#B36>
- Raja, S., S., P. D., & K., S. R. (2011). Web Based Decision Support System for Kidney. *International Journal of Computer Applications* (0975 – 8887), 9.
- Roth, A. E., Sonmez, T., & Unver, M. U. (2005). Kidney Exchange. *Quarterly Journal of Economics*, 32.
- 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>
- Sedgewick, R., & Wayne, K. (2020). Directed Graphs. In R. Sedgewick, & K. Wayne, *Algorithms, 4th Edition* (p. 955). New Jersey: Princeton University.

Tullis, T., & Albert, B. (2013, June 3). *Performance Metrics*. Retrieved from sciencedirect.com:

<https://www.sciencedirect.com/science/article/pii/B978012415781100004>

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#:~:text=Sebenarnya%2C%20jumlah%20pasien%20yang%20memerlukan,orang%20yang%20memerlukan%20transplantasi%20ginjal>

hambatannya#:~:text=Sebenarnya%2C%20jumlah%20pasien%20yang%20memerlukan,orang%20yang%20memerlukan%20transplantasi%20ginjal.

LAMPIRAN A. Kode Sumber Algoritma Pencarian Pemetaan Kecocokan

1. Kelas ExchangeAlgorithm

```
from .util.graph_vis import GraphVisualization
from .util.flatten import flatten

class ExchangeAlgorithm(object):
    # Base Class for Exchange Algorithms
    def __init__(self):
        self.cycles = []
        self.vertices = []

    @staticmethod
    def element_in_assigned(cycle, assigned):
        return any(item in cycle for item in assigned)

    def get_num_of_matched_pairs(self):
        return len(flatten(self.cycles))

    def occuring_cycle_removal(self, cycles):
        # remove cycles with previous occurring vertices
        assigned = set()
        for cycle in cycles:
            if not ExchangeAlgorithm.element_in_assigned(cycle,
assigned):
                self.cycles.append(cycle)

                # add index to assigned
                for index in cycle:
                    assigned.add(index)

    # Graph Visualization Methods
    def show_donation_mapping_graph(self):
        gv = GraphVisualization()

        # Add vertices
        gv.add_vertices_from(self.vertices)

        # to visualize cycle, add start node as end node
        vis_cycles = [cycle + cycle[:1] for cycle in self.cycles]
        edges = [ed for cycle in vis_cycles for ed in zip(cycle[:-
1], cycle[1:])]
        gv.add_edges_from(edges)
        gv.visualize('rb')
```

```

def show_donation_mapping_text(self):
    for cycle in self.cycles:
        print(" --> ".join(map(str, cycle)))

def show_donation_mapping(self, print_method='text'):
    # print_method:
    #   - "text": print donation mapping as text
    #   - "graph": print reduced graph of donation mapping
    if (print_method == 'text'):
        self.show_donation_mapping_text()
    elif (print_method == 'graph'):
        self.show_donation_mapping_graph()
    else:
        raise ValueError("Print method not found, available
methods: 'text', 'graph'")

```

2. Kelas EdmondsAlgorithm

```

from .exchange_algorithm import ExchangeAlgorithm
from .util.sort_cycle_by_priority import edmond_priority_cycle

class EdmondsAlgorithm(ExchangeAlgorithm):
    # benchmark algorithm: Edmond's
    # Inherits from Exchange Algorithm
    def __init__(self, priority_threshold=10):
        super().__init__()

        # add priority_threshold
        #   get cycles only if priority in threshold tolerance
        #   delete otherwise
        self.priority_threshold = priority_threshold

    # Method to finalize exchange of directed graphs
    def finalize_exchange(self, directed_graph):
        cycles = edmond_priority_cycle(directed_graph,
self.priority_threshold)
        # Edmond's Algorithm only solves 2 way exchange
        cycles = [cycle for cycle in cycles if (len(cycle) == 2)]

        self.occuring_cycle_removal(cycles)
        self.vertices = directed_graph.get_vertices()

```

3. Kelas FirstAcceptNWay

```
from .exchange_algorithm import ExchangeAlgorithm

class FirstAcceptNWay(ExchangeAlgorithm):
    # orientation: first come first serve
    # Inherits from Exchange Algorithm
    def __init__(self, n, method):
        super().__init__()

        self.n = n
        # method:
        # - "maximum": n as maximum number of edge in cycle
        # - "exact": n as exact number of edge in cycle
        if (method == 'maximum'):
            self.method = self.maximum_method
        elif (method == 'exact'):
            self.method = self.exact_method
        else:
            raise ValueError("Method not found, available methods:
'maximum', 'exact'")

    def maximum_method(self, cycles):
        return [cycle for cycle in cycles if (len(cycle) <= self.n)]

    def exact_method(self, cycles):
        return [cycle for cycle in cycles if (len(cycle) == self.n)]

    # Method to finalize exchange of directed graphs
    def finalize_exchange(self, directed_graph):
        # prune if len > n
        cycles = self.method(directed_graph.get_cycles())

        self.occuring_cycle_removal(cycles)
        self.vertices = directed_graph.get_vertices()
```

4. Kelas PriorityBasedNWay

```
from .exchange_algorithm import ExchangeAlgorithm
from .util.sort_cycle_by_priority import greedy_priority,
infrequent_priority

class PriorityBasedNWay(ExchangeAlgorithm):
    # orientation: sort cycles by priority
    # Inherits from Exchange Algorithm
```

```

def __init__(self, n, method, priority='greedy'):
    super().__init__()

    self.n = n
    # method:
    # - "maximum": n as maximum number of edge in cycle
    # - "exact": n as exact number of edge in cycle
    if (method == 'maximum'):
        self.method = self.maximum_method
    elif (method == 'exact'):
        self.method = self.exact_method
    else:
        error_message = "Method not found, available methods:
'maximum', 'exact'"
        raise ValueError(error_message)

    # priority -> which gets the higher priority?
    # - "greedy": cycles with most amount of matches
    # - "infrequent": cycles with vertices that has least
amount of edges
    if (priority == 'greedy'):
        self.priority = greedy_priority
    elif (priority == 'infrequent'):
        self.priority = infrequent_priority
    else:
        error_message = "Priority Method not found, available
priorities: 'greedy', 'infrequent'"
        raise ValueError(error_message)

def maximum_method(self, cycles):
    return [cycle for cycle in cycles if (len(cycle) <= self.n)]

def exact_method(self, cycles):
    return [cycle for cycle in cycles if (len(cycle) == self.n)]

# Method to finalize exchange of directed graphs
def finalize_exchange(self, directed_graph):
    # sort cycles by priority
    sorted_cycles = self.priority(directed_graph)

    # prune if len > n
    cycles = self.method(sorted_cycles)

    self.occuring_cycle_removal(cycles)
    self.vertices = directed_graph.get_vertices()

```


LAMPIRAN B. Hasil Eksperimen Perhitungan *Matching Efficiency* untuk Kombinasi Algoritma dan Parameter yang berbeda-beda

Data Seed	Baseline: Edmond	First Accept(N=3, Method=Max)	First Accept(N=10, Method=Max)	First Accept(N=3, Method=Exact)	First Accept(N=10, Method=Exact)	Priority Based(N=3, Method=Max, Priority=Greedy)	Priority Based(N=10, Method=Max, Priority=Greedy)	Priority Based(N=3, Method=Max, Priority=Infrequent)	Priority Based(N=10, Method=Max, Priority=Infrequent)	Priority Based(N=3, Method=Exact, Priority=Greedy)	Priority Based(N=10, Method=Exact, Priority=Greedy)	Priority Based(N=3, Method=Exact, Priority=Infrequent)	Priority Based(N=10, Method=Exact, Priority=Infrequent)
0	36	43	43	39	40	41	45	39	39	39	40	39	30
1	34	41	44	30	40	38	45	41	43	30	40	30	40
2	36	44	45	33	40	45	44	43	44	33	40	36	30
3	46	53	61	51	60	57	62	53	54	51	60	48	50
4	44	56	59	48	50	54	60	54	56	48	50	45	50
5	28	32	32	33	30	33	35	28	28	33	30	33	30
6	36	40	42	33	30	41	42	40	40	33	30	33	20
7	48	57	61	51	50	59	63	57	58	51	50	51	50
8	46	51	51	36	40	52	56	48	48	36	40	39	40
9	50	56	56	39	40	55	57	58	58	39	40	39	40
10	48	55	55	45	50	53	55	55	54	45	50	45	40
11	50	56	57	36	50	52	60	57	56	36	50	36	50
12	50	53	54	39	50	53	57	52	51	39	50	42	40
13	48	57	57	39	50	51	57	54	55	39	50	39	50
14	38	50	51	39	30	51	53	48	50	39	30	39	40
15	48	56	59	45	50	57	58	54	54	45	50	45	40
16	38	46	46	39	40	43	48	43	45	39	40	39	40
17	34	40	41	27	30	39	44	37	38	27	30	27	30
18	48	52	53	36	50	52	53	52	52	36	50	39	50
19	40	47	49	39	40	45	50	46	46	39	40	39	30
20	40	51	53	42	50	50	55	47	50	42	50	42	40
21	34	48	50	45	40	49	48	47	47	45	40	42	30
22	42	49	50	39	40	49	53	47	50	39	40	42	40
23	42	44	47	36	50	44	53	44	45	36	50	36	40
24	36	41	42	33	40	41	46	42	41	33	40	33	30
25	40	50	50	36	40	48	50	49	49	36	40	42	40
26	48	59	60	48	50	58	63	54	56	48	50	45	40
27	36	43	44	27	30	43	46	37	37	27	30	27	20
28	40	52	53	45	50	51	53	45	45	45	50	45	30
29	44	46	47	36	40	46	55	47	54	36	40	36	40

LAMPIRAN C. Hasil Eksperimen Rasio Perbandingan Algoritma N arah dengan Algoritma *Baseline*

Data Seed	Baseline: Edmond	First Accept(N=3, Method=Max)	First Accept(N=10, Method=Max)	First Accept(N=3, Method=Exact)	First Accept(N=10, Method=Exact)	Priority Based(N=3, Method=Max, Priority=Greedy)	Priority Based(N=10, Method=Max, Priority=Greedy)	Priority Based(N=3, Method=Max, Priority=Infrequent)	Priority Based(N=10, Method=Max, Priority=Infrequent)	Priority Based(N=3, Method=Exact, Priority=Greedy)	Priority Based(N=10, Method=Exact, Priority=Greedy)	Priority Based(N=3, Method=Exact, Priority=Infrequent)	Priority Based(N=10, Method=Exact, Priority=Infrequent)
0	1.00	1.19	1.19	1.08	1.11	1.14	1.25	1.08	1.08	1.08	1.11	1.08	0.83
1	1.00	1.21	1.29	0.88	1.18	1.12	1.32	1.21	1.26	0.88	1.18	0.88	1.18
2	1.00	1.22	1.25	0.92	1.11	1.25	1.22	1.19	1.22	0.92	1.11	1.00	0.83
3	1.00	1.15	1.33	1.11	1.30	1.24	1.35	1.15	1.17	1.11	1.30	1.04	1.09
4	1.00	1.27	1.34	1.09	1.14	1.23	1.36	1.23	1.27	1.09	1.14	1.02	1.14
5	1.00	1.14	1.14	1.18	1.07	1.18	1.25	1.00	1.00	1.18	1.07	1.18	1.07
6	1.00	1.11	1.17	0.92	0.83	1.14	1.17	1.11	1.11	0.92	0.83	0.92	0.56
7	1.00	1.19	1.27	1.06	1.04	1.23	1.31	1.19	1.21	1.06	1.04	1.06	1.04
8	1.00	1.11	1.11	0.78	0.87	1.13	1.22	1.04	1.04	0.78	0.87	0.85	0.87
9	1.00	1.12	1.12	0.78	0.80	1.10	1.14	1.16	1.16	0.78	0.80	0.78	0.80
10	1.00	1.15	1.15	0.94	1.04	1.10	1.15	1.15	1.13	0.94	1.04	0.94	0.83
11	1.00	1.12	1.14	0.72	1.00	1.04	1.20	1.14	1.12	0.72	1.00	0.72	1.00
12	1.00	1.06	1.08	0.78	1.00	1.06	1.14	1.04	1.02	0.78	1.00	0.84	0.80
13	1.00	1.19	1.19	0.81	1.04	1.06	1.19	1.13	1.15	0.81	1.04	0.81	1.04
14	1.00	1.32	1.34	1.03	0.79	1.34	1.39	1.26	1.32	1.03	0.79	1.03	1.05
15	1.00	1.17	1.23	0.94	1.04	1.19	1.21	1.13	1.13	0.94	1.04	0.94	0.83
16	1.00	1.21	1.21	1.03	1.05	1.13	1.26	1.13	1.18	1.03	1.05	1.03	1.05
17	1.00	1.18	1.21	0.79	0.88	1.15	1.29	1.09	1.12	0.79	0.88	0.79	0.88
18	1.00	1.08	1.10	0.75	1.04	1.08	1.10	1.08	1.08	0.75	1.04	0.81	1.04
19	1.00	1.18	1.23	0.98	1.00	1.13	1.25	1.15	1.15	0.98	1.00	0.98	0.75
20	1.00	1.28	1.33	1.05	1.25	1.25	1.38	1.18	1.25	1.05	1.25	1.05	1.00
21	1.00	1.41	1.47	1.32	1.18	1.44	1.41	1.38	1.38	1.32	1.18	1.24	0.88
22	1.00	1.17	1.19	0.93	0.95	1.17	1.26	1.12	1.19	0.93	0.95	1.00	0.95
23	1.00	1.05	1.12	0.86	1.19	1.05	1.26	1.05	1.07	0.86	1.19	0.86	0.95
24	1.00	1.14	1.17	0.92	1.11	1.14	1.28	1.17	1.14	0.92	1.11	0.92	0.83
25	1.00	1.25	1.25	0.90	1.00	1.20	1.25	1.23	1.23	0.90	1.00	1.05	1.00
26	1.00	1.23	1.25	1.00	1.04	1.21	1.31	1.13	1.17	1.00	1.04	0.94	0.83
27	1.00	1.19	1.22	0.75	0.83	1.19	1.28	1.03	1.03	0.75	0.83	0.75	0.56
28	1.00	1.30	1.33	1.13	1.25	1.28	1.33	1.13	1.13	1.13	1.25	1.13	0.75
29	1.00	1.05	1.07	0.82	0.91	1.05	1.25	1.07	1.23	0.82	0.91	0.82	0.91