

Rancang Bangun Aplikasi *Chatbot* Rekomendasi Jadwal Penggunaan Ruang Berbasis *Graph Coloring* Menggunakan Dialogflow Dan Neo4j

Diaz Ardian Alvianto¹, Ricky Eka Putra²,

^{1,2} Jurusan Teknik Informatika/ S1 Teknik Informatika, Universitas Negeri Surabaya

¹diazalvianto@mhs.unesa.ac.id

²rickyeka@unesa.ac.id

Abstrak—Peningkatan jumlah mahasiswa Universitas Negeri Surabaya pada Program Studi Teknik Informatika merupakan hal yang tidak bisa dihindari. Hal itu berpengaruh pada banyaknya jadwal perkuliahan yang harus diatur dengan baik. Salah satu variabel penting dalam menyusun penjadwalan yaitu ketersediaan ruangan perkuliahan. Sehingga dibutuhkan suatu sistem yang dapat membantu manajemen ruang untuk mengatur antara jadwal kuliah dengan ruangan yang tersedia. Sistem yang dirancang berupa aplikasi *chatbot* rekomendasi jadwal penggunaan ruangan berbasis *Graph Coloring* menggunakan Dialogflow dan Neo4j. Penelitian ini mengaplikasikan *graph coloring* algoritma *welch-powell* untuk membuat rekomendasi penggunaan ruang kelas yang efektif. Hasil penjadwalan ruangan yang diperoleh nantinya akan dimasukkan kedalam Neo4j sebagai database aplikasi. *Chatbot* digunakan untuk mengimplementasikan hasil dari penjadwalan ruangan, sehingga akan menampilkan *output* rekomendasi ruangan tersedia dalam aplikasi *chatbot*. Algoritma yang diterapkan mampu menghasilkan ketepatan hasil untuk pemesanan ruang kelas pengganti dan ruang sidang dari *chatbot*, yaitu pada pengujian menggunakan algoritma genetika untuk membuat jadwal perkuliahan menghasilkan tidak ada bentrok antar kelas dan dosen dari *best fitness* dengan besar probabilitas *crossover* 0,75, probabilitas mutasi: 0,50, dan kromosom/ populasi 100. Sedangkan pada pengujian *graph coloring* menggunakan fitur K-1 Coloring Algorithm pada Neo4j mengasilkan pemetaan *node* dan *edges* untuk rekomendasi penggunaan ruang kelas. Pada fitur pemesanan ruang kelas pengganti dilakukan 20 kali percobaan melihat rekomendasi kelas dari *chatbot* dan menghasilkan balasan yang sudah sesuai dengan aspek menurut hari, ketersediaan dosen dan kelas. Hal tersebut menunjukkan bahwa aplikasi *chatbot* yang telah dibuat telah berhasil dan layak diterapkan untuk rekomendasi jadwal penggunaan ruangan.

Kata Kunci— Penjadwalan, *Chatbot*, *Graph Coloring*, DialogFlow, Neo4j

I. PENDAHULUAN

Penjadwalan merupakan sebuah aktivitas perencanaan menentukan waktu di mana setiap kegiatan akan dilakukan secara keseluruhan. Menurut Baker penjadwalan adalah proses untuk melakukan tugas dengan menggunakan sumber-sumber yang tersedia pada waktu yang telah ditetapkan[1]. Sedangkan menurut Blazewicz penjadwalan adalah membangun penentuan waktu penggunaan dari peralatan, fasilitas dan aktivitas manusia dalam suatu organisasi[2]. Untuk membuat sebuah penjadwalan tentunya tidak mudah, terdapat banyak tantangan

dan faktor-faktor dalam proses penjadwalan yang harus dipertimbangkan. Kasus yang sering terjadi di Universitas adalah menentukan jadwal mata kuliah dan ruang perkuliahan. Permasalahan yang sering ditemui yaitu menentukan penjadwalan penggunaan ruangan karena faktor ruangan yang minim atau perubahan jadwal mata kuliah yang mendadak, sehingga terjadi sebuah kasus dimana terpakainya ruangan yang tidak seharusnya dipakai oleh kelas tertentu atau saling memperebutkan ruangan.

Berdasarkan pengamatan penelitian menemukan bahwa *graph coloring* salah satu teknik yang dapat digunakan dalam penyelesaian permasalahan tersebut. *Graph coloring* merupakan metode untuk menentukan jumlah warna minimum diberikan pada *nodes* atau *edges* dalam sebuah graf sehingga *nodes* atau *edges* yang berhubungan tidak memiliki warna yang sama, metode ini dapat dipastikan meminimalkan atau sampai menghilangkan pelanggaran *constraint*. Pada penelitian yang dilakukan oleh Ganguli, hasil pemberian pewarnaan graf pada beberapa set data dengan mempertimbangkan berbagai kendala untuk mengoptimalkan solusi jadwal kuliah yang bebas konflik sesuai kondisi preferensial maksimum[3]. Sedangkan pada penelitian yang dilakukan oleh Nandhini, hasil nilai dari algoritma diklaim mencapai kinerja yang hampir optimal dalam waktu polinomial[4]. Pada penelitian yang dilakukan oleh Eka, penyusunan jadwal dilakukan dengan memetakan jadwal ke dalam graf dan menentukan bilangan kromatik graf menggunakan algoritma *graph coloring* untuk menghindari bentrokan jadwal[5].

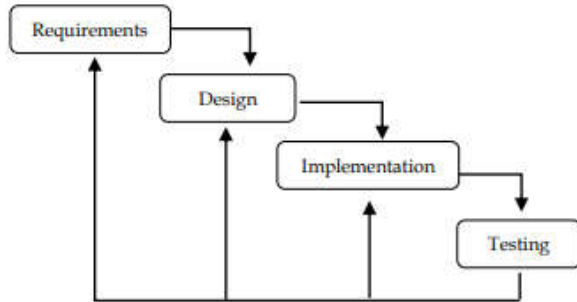
Implementasi metode *graph coloring* dibutuhkan sebuah antarmuka yang bisa secara intuitif dapat mengenali input dari pengguna. Dalam hal ini penggunaan Telegram sebagai antarmuka yang cocok dikarenakan fitur Telegram yang banyak dan juga *open source*. Walaupun telegram menggunakan metode enkripsi yang transparan namun sistem privasi Telegram sangat baik. Untuk *chatbot*, Telegram memperbolehkan pihak ketiga untuk menggunakan *bot* di *platform* mereka. Pihak ketiga bisa menggunakan *API* dan dokumentasi dari Telegram untuk mengembangkan berbagai macam *bot*. DialogFlow juga dibutuhkan untuk *Natural Language Processing* atau NLP yang mana berguna mengolah perintah dari *user* untuk *Training* otomatis di sistem DialogFlow yang dibutuhkan untuk balasan dari respon *chatbot*. Karena penjadwalan adalah hal yang penting terhadap terciptanya kegiatan belajar mengajar untuk berjalan dengan

baik di Universitas, maka salah satu upaya untuk memudahkan penjadwalan dengan tepat dan akurat adalah dengan menggunakan metode yang diterapkan pada aplikasi yang tepat dan fleksibel dalam penggunaannya.

II. METODOLOGI PENELITIAN

Berikut ini merupakan alur penelitian untuk membuat rancang bangun aplikasi *chatbot* rekomendasi jadwal penggunaan ruangan berbasis dialogflow dan neo4j secara umum dapat dilihat pada Gambar 1.

A. Rancangan Penelitian



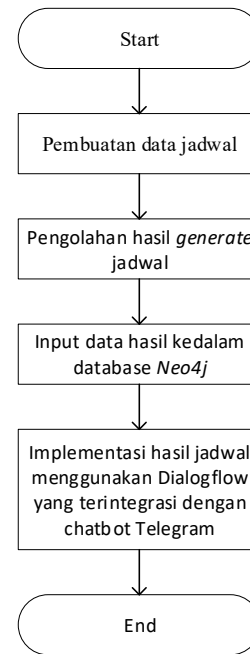
Gambar. 1 Alur Penelitian

Gambar 1 menggambarkan tentang alur penelitian rancang bangun aplikasi *chatbot* rekomendasi jadwal penggunaan ruangan berbasis dialogflow dan neo4j. Adapun tahapan yang dilakukan pada penelitian ini yaitu analisis kebutuhan dengan melakukan studi pustaka untuk mencari literatur yang relevan yang dapat menjadi referensi pendukung dalam penelitian. Literatur yang berhubungan dengan kecerdasan buatan, *database* dan penerapan *graph coloring*. Kemudian membuat desain yang akan dibangun seperti pada *chatbot* telegram pada umumnya. Pada tahap implementasi, penelitian ini menggunakan bahasa python untuk mengolah *script code* dan akan diimplementasikan dalam bentuk *chatbot* telegram. Pada tahap pengujian sistem dilakukan untuk mengetahui ketepatan dialogflow dalam mengeksekusi perintah. Kemudian menguji ketepatan hasil penjadwalan dari aplikasi *chatbot* dan juga mengetahui hasil *response time chatbot* melalui *proxy server* ngrok.

B. Desain

1. Desain Alur Pembuatan Program

Dalam penelitian ini memiliki alur pembuatan program, yang dijelaskan pada *flow chart* pada Gambar 2 sebagai berikut:

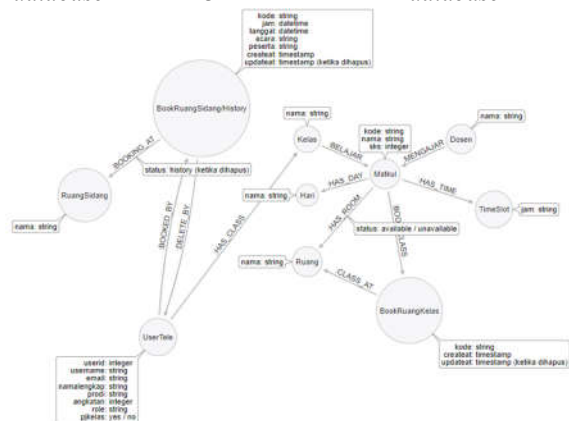


Gambar. 2 Alur Pembuatan Program

Gambar 2 merupakan alur proses pembuatan program dalam penelitian ini. Langkah yang dilakukan adalah membuat data jadwal dengan menggunakan algoritma genetika dan *graph coloring*. Kemudian data hasil *generate* diolah menjadi data jadwal selama satu minggu. Selanjutnya data yang sudah diolah dimasukkan kedalam *database* Neo4j sebagai *database* sistem *chatbot*. Implementasi dari hasil jadwal menggunakan *chatbot* dengan bantuan dialogflow sebagai NLP, yang terintegrasi dengan *chatbot* Telegram.

2. Desain Database Neo4j

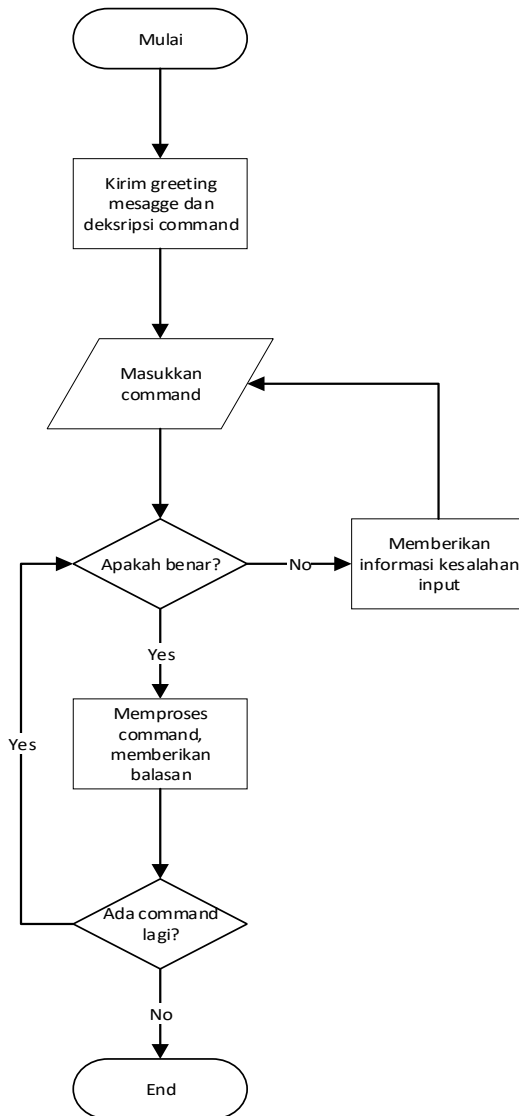
Pada penelitian ini menggunakan Neo4j sebagai *database*. Gambar 3 adalah desain dari *database*.



Gambar. 3 Desain database Neo4j

3. Desain Sistem Chatbot

Dalam sistem ini menggunakan aplikasi Telegram sebagai *user interface* untuk menampilkan hasil dari proses pencarian pilihan ruangan dengan metode *graph coloring*. Alur sistem yang akan dibuat dijelaskan pada *flow chart* pada Gambar 4 sebagai berikut:



Gambar. 4 Alur Desain Sistem Chatbot

Pada Gambar 4 menampilkan alur desain sistem rancang bangun aplikasi *chatbot* rekomendasi jadwal penggunaan ruangan berbasis dialogflow dan neo4j secara umum. Adapun langkah yang dilakukan pada penelitian ini yaitu dengan memasukkan *command* “/Start” untuk memulai percakapan dengan *bot* pada Telegram. Kemudian input *command* untuk memulai fitur yang diinginkan oleh pengguna.

Tahap selanjutnya *bot* akan memproses *keyword* yang diinput dengan mengecek input yang diberikan pengguna apakah benar atau salah. Jika pada proses *input keyword* tidak menemukan hasil balasan atau salah, maka *bot* akan meminta pengguna untuk memasukkan *command* kembali, dan jika proses benar *bot* akan memberikan balasan. Apabila ada *command* lain, maka proses akan kembali ke pengecekan *keyword*.

C. Algoritma Genetika

Algoritma genetika adalah sebuah algoritma untuk penyelesaian masalah ataupun optimasi. Algoritma genetika secara khusus diimplementasikan sebagai simulasi komputer, dimana kromosom atau sebuah populasi representasi dari solusi calon individu pada suatu masalah optimasi berkembang menjadi solusi yang semakin lebih baik. Inisialisasi populasi akan diselesaikan secara acak oleh algoritma dan diperbaiki melalui pengulangan dengan operator mutasi, persilangan atau *crossover*, dan seleksi[6][7][8]. Operator genetika dalam penelitian ini terdiri dari:

1. Pemilihan Individu Induk

Algoritma genetika memiliki 2 (dua) cara untuk seleksi individu induk, dengan cara *rank based fitness* atau *roulette wheel selection*. Individu induk yang sudah terpilih adalah individu induk yang sudah melalui berbagai macam proses. Untuk mendapatkan individu induk terbaik dapat menggunakan *roulette wheel selection*. Untuk menggabungkan fungsi *fitness* dan *roulette wheel selection*, maka fungsi *fitness* yang terbentuk digunakan sebagai probabilitas dan kromosom[6][7].

$$probabilitas\ kromosom = \frac{1}{fitness}$$

2. Crossover

Crossover atau bisa disebut juga kawin silang adalah dimana algoritma genetika menggabungkan dua kromosom *parent* menjadi kromosom baru. Penelitian ini menggunakan *one-cut point* atau memilih acak gen dalam kromosom induk[6][7].

3. Mutasi

Mutasi dilakukan saat proses pembentukan kromosom induk sebelum proses *crossover*. Proses mutasi dilakukan untuk memeriksa dosen pengajar pada slot yang sama[6][7].

D. Graph Coloring

Metode *Graph Coloring* digunakan untuk menentukan jumlah warna minimum yang dapat diberikan pada *nodes* atau *edges* di dalam sebuah *graph* sehingga *nodes* atau *edges* yang saling berhubungan tidak memiliki warna yang sama. Berdasarkan cara kerjanya tersebut, metode ini dapat

dipastikan menghasilkan penjadwalan kegiatan dapat meminimalkan atau sampai menghilangkan pelanggaran terhadap *constraints*[3][12].

E. Dialogflow

Dialogflow adalah platform pemrosesan bahasa alami (NLP) yang dapat digunakan untuk membangun aplikasi dan pengalaman percakapan untuk pelanggan perusahaan dalam berbagai bahasa dan pada berbagai *platform*. Produk yang dikembangkan Google ini memungkinkan para pengembang untuk membuat antarmuka berbasis teks dan percakapan suara untuk menjawab pertanyaan pelanggan dalam berbagai bahasa [9].

F. Graph Database Neo4j

Database adalah model penyimpanan data berbasis teori graf dimana data dilambangkan sebagai *Node* dan penghubung setiap *Node* adalah *edges*. *Graph Database* termasuk dalam NoSQL (*Non Relational / Not Only SQL*). NoSQL merubah struktur penyimpanan data konvensional bertujuan untuk mendukung data-data yang tidak terstruktur, *scalable (Big Data)*, pencarian cepat, dan ekstraksi informasi tersembunyi [10].

G. Chatbot

Chatter bot atau bisa juga disebut *chatbot* adalah program komputer untuk menyimulasikan percakapan intelektual dengan satu atau lebih manusia secara audio maupun teks. *Chatbot* dikategorikan sebagai kecerdasan buatan atau *artificial intelligence*. Yang membedakan chatbot dengan sistem pemrosesan Bahasa alami atau *Natural Language Processing* adalah sederhananya algoritma yang digunakan. Meskipun banyak *bot* yang dapat menanggapi manusia, sebenarnya *bot* tersebut hanya memahami kata kunci dalam setiap input dan membalasnya dengan kata kunci yang cocok atau pola kalimat yang mirip [11][9].

H. Pengujian sistem

Pada penelitian ini dilakukan pengujian terhadap sistem yang telah dikerjakan. Pengujian sistem dilakukan dengan tujuan untuk mengetahui keberhasilan dari sistem. Terdapat dua pengujian yang akan penulis lakukan yaitu sebagai berikut :

1. Ketepatan Dialogflow dalam mengeksekusi perintah dari *user* atau pengguna aplikasi *chatbot*. Pengujian dilakukan dengan melihat keakuratan *action* dan *parameter* apakah telah sesuai dengan permintaan kata kunci yang dimasukkan oleh *user* atau pengguna aplikasi *chatbot* tersebut.
2. Ketepatan hasil rekomendasi ruangan tersedia dalam aplikasi *chatbot*, data hasil *generate* jadwal yang diambil dari *database* sistem akan dilihat dan dicocokkan apakah sudah sesuai dengan perintah atau kata kunci yang dimasukkan oleh *user* atau pengguna aplikasi *chatbot*. Pada pengujian ini akan dilakukan dengan perintah kata kunci *user* yang

beragam untuk melihat keakuratan hasil kueri *database*.

3. Performa kecepatan *reply* respon *chatbot* melalui *proxy server* ngrok. Pengujian dilakukan untuk menentukan seberapa cepat *server* dalam memberikan *reply* dari masukan atau permintaan yang diminta oleh *user*, selain itu juga untuk menentukan seberapa besar *delay time server* dalam proses ngrok tersebut.

III. HASIL DAN PEMBAHASAN

Pada bagian ini menjelaskan hasil pengujian terhadap implementasi yang telah dilakukan beserta analisisnya. Pengujian dilakukan untuk menilai apakah seluruh kebutuhan dan analisis yang telah dispesifikasikan sebelumnya telah terpenuhi. Rincian yang diujikan mulai dari ketepatan hasil yang diminta pengguna untuk pesan ruang kelas dan ruang sidang dari chatbot.

A. Hasil *generate* jadwal kuliah menggunakan algoritma genetika dan *graph coloring*.

1. Analisis Data

Pada pengujian menggunakan algoritma genetika, pengujian dilakukan menggunakan bahasa python yang dijalankan pada komputer dengan spesifikasi, AMD A9-9400 RADEON R5, 5 COMPUTE CORES 2C+3G (2CPU) 2.4GHz dan 4GB RAM dengan studi kasus penjadwalan pada Jurusan Teknik Informatika Universitas Negeri Surabaya. Sampel studi kasus yang digunakan adalah data pada semester gasal 2019/2020. Sedangkan pada pengujian *graph coloring* menggunakan fitur K-1 Coloring Algorithm pada Neo4j.

2. Algoritma Genetika

Pada proses pengujian menggunakan algoritma genetika, uji coba dilakukan sebanyak 5 kali setiap nilai dengan banyaknya jumlah kromosom/ populasi adalah 20,40,60,80 dan 100; probabilitas *crossover* 0,25, 0,35, 0,45, 0,60 dan 0,75; probabilitas mutasi 0,10, 0,20, 0,30, 0,40 dan 0,50; dan banyaknya jumlah generasi/ iterasi 10000. Hasil dari pengujian ini seperti pada Tabel 1 dan Gambar 5.

Tabel 1 Hasil Best Fitness

Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5
Fitness: 0.3258333	Fitness: 0.3354166	Fitness: 0.2800396	Fitness: 0.2296974	Fitness: 0.2075995
Crossover : 0.25	Crossover : 0.35	Crossover : 0.45	Crossover : 0.60	Crossover : 0.75
Mutasi : 0.10	Mutasi : 0.20	Mutasi : 0.30	Mutasi : 0.40	Mutasi : 0.50
Populasi : 20	Populasi : 40	Populasi : 60	Populasi : 80	Populasi : 100



Gambar. 5 Grafik best fitness

Dari hasil *best fitness* yang ditunjukkan pada Tabel 1 dan Gambar 5, bisa diambil kesimpulan bahwa percobaan ke 5 dengan besar probabilitas crossover 0,75, probabilitas mutasi 0,50, dan kromosom/ populasi 100 adalah fitness yang terbaik dari beberapa uji coba yang lain.

3. K-1 Coloring Algorithm/ Graph Coloring.

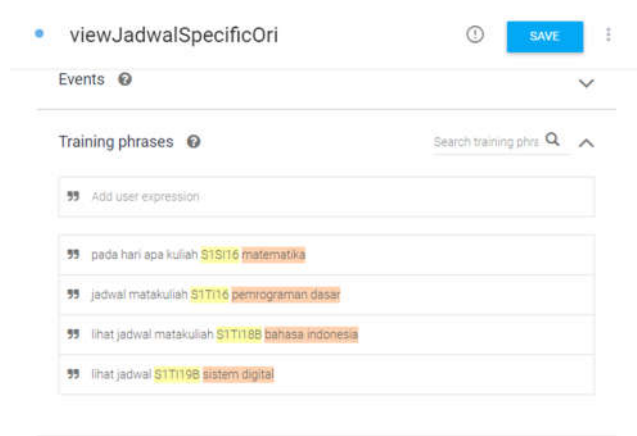
Pada proses ini, hasil dari *generate* jadwal yang diperoleh menggunakan algoritma genetika akan diolah menggunakan K-1 *Coloring Algorithm* Neo4j yang berbasis algoritma *Welch - Powell* untuk menentukan rekomendasi ruang kelas. Gambar 6 adalah data yang sudah diolah oleh K1-Coloring Algorithm Neo4j.



Gambar. 6 Node K1 – Coloring Alorithm

B. Hasil Ketepatan Respon Chatbot

Ketepatan respon dari *chatbot* dapat diukur dari *keyword user* dengan parameter yang ada pada Dialogflow. Dialogflow mempelajari semua *keyword user says* yang sudah dimasukkan dalam *intens* dan akan dicek terdapat *action* pada *intens* yang terdapat *keyword user says*. Gambaran dari pengecekan ketepatan respon terdapat pada Gambar 7.



Gambar. 7 Training phrases DialogFlow

Dalam *intens* Dialogflow, diperlukan sebuah *action* dan *parameter* untuk menentukan *user says* sudah berada pada *intens* yang tepat atau tidak. Gambar 8 adalah contoh *action* dan *parameter intens*.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Kelas	@Kelas	\$Kelas	<input type="checkbox"/>
<input type="checkbox"/>	matkul	@matkul	\$matkul	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

Gambar. 8 Action and parameters DialogFlow

Data *intens* dikirimkan ke *webhook* untuk meneruskan perintah ke sistem *chatbot*. Pada Gambar 9, *user says* akan mengenali *action* yang sudah ditetapkan, lalu sistem *chatbot* akan memberikan balasan yang sesuai. Jika balasan *error* atau *chatbot* tidak tahu, ada kesalahan dalam penempatan *action* pada *user says*.

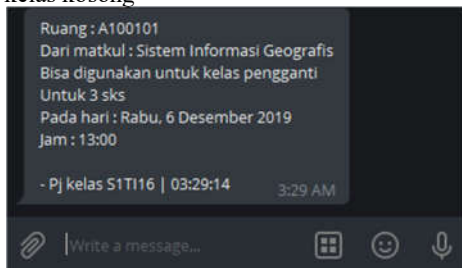


Gambar. 9 Data JSON Ketepatan Respon Chatbot

C. Hasil fitur chatbot.

1. Report kelas tidak dipakai dari jadwal aktif.

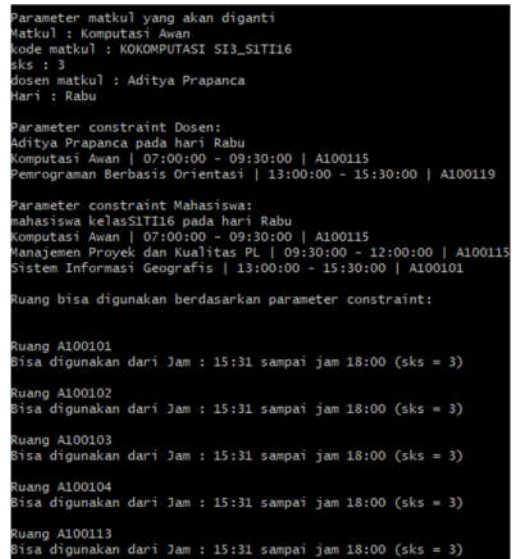
Fitur ini memungkinkan untuk pj kelas dapat melaporkan kepada pj kelas lain bahwa ada kelas kosong yang dapat dipakai. Pj kelas hanya perlu menuliskan matkul yang akan kosong, tanggal untuk kelas itu kosong dan kelas akan direkomendasikan ke kelas lain. Gambar 10 adalah contoh hasil dari report ruang kelas kosong



Gambar. 10 Hasil dari report

2. Rekomendasi ruang kelas tersedia.

Pada fitur ini, user dapat mencari kelas yang dapat digunakan sebagai kelas pengganti. Dengan fitur ini, bot memberikan rekomendasi ruangan yang bisa digunakan sesuai dengan jumlah sks/ timeslot, pertimbangan jadwal dosen dan pertimbangan jadwal kelas. Rekomendasi ruang kelas juga memungkinkan dari kelas yang sudah melaporkan bahwa matakuliah tidak diadakan pada waktu itu pada sistem chatbot.



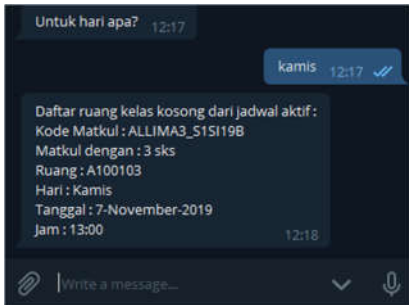
Gambar. 11 Uji coba rekomendasi ruang kelas

Gambar 11 adalah hasil uji coba dalam sistem python berdasarkan ruangan yang tidak boleh bentrok antara jadwal dosen, kelas, dan ruang yang sudah dipakai. Dari uji coba diatas menunjukkan rekomendasi kelas sudah sesuai dengan mata kuliah yang akan diganti yaitu berdasarkan hari dan sks. Lalu akan dicocokkan dengan jadwal mengajar dosen pada hari tersebut dan jadwal kelas pada hari tersebut. Maka ditemukan ruangan pengganti yang cocok dengan keterangan sks yang sesuai.



Gambar. 12 Rekomendasi ruang kelas

Pada Gambar 12, bisa dilihat rekomendasi kelas yang dapat digunakan berdasarkan batasan dari jadwal kelas dan jadwal dosen pada *chatbot* telegram.



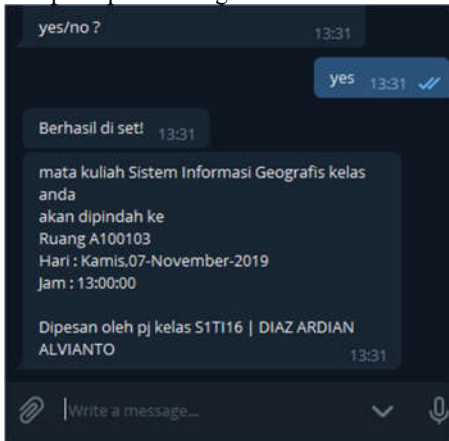
Gambar. 13 Rekomendasi ruang kelas jadwal aktif

Pada Gambar 13, hasil dari ruang kelas kosong berdasarkan jadwal aktif perkuliahan.

3. Pesan ruang kelas pengganti.

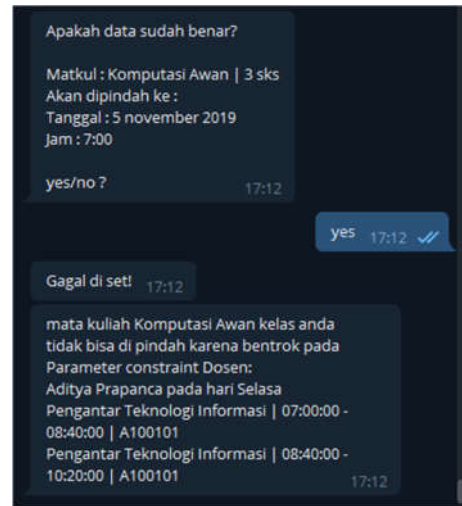
Setelah melihat rekomendasi kelas yang ada, pengguna dapat memesan ruang untuk pengganti perkuliahan.

a. Tampilan pesan ruang kelas



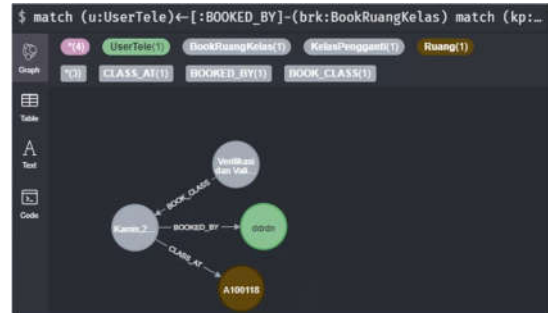
Gambar. 14 Tampilan pesan ruang kelas

Gambar 14 merupakan tampilan pesan ruang kelas pengganti pada *chatbot* telegram. *User* diminta untuk memasukkan kode ruang kelas kosong jika dari rekomendasi kelas aktif, mata kuliah yang ingin diganti, dan tanggal kapan mata kuliah kosong tersebut. Dicocokkan dengan data *database* dan sistem python apakah sudah memenuhi syarat, jika sukses maka akan muncul keterangan dan pesan otomatis dikirim ke seluruh anggota kelas yang sudah mendaftar bot. Jika gagal maka ada bentrok dari jadwal dosen atau jadwal kelas tersebut seperti pada contoh Gambar 15.



Gambar. 15 Bentrok pesan ruang kelas

b. Tampilan *record* pesanan ruang kelas pada Neo4j



Gambar. 16 *Record* database pesan ruang kelas

Gambar 16 merupakan tampilan dari *record* pesan ruang kelas pengganti. *User* atau *pj* kelas membuat pesanan ruang pengganti dan sudah tersimpan di *database*.

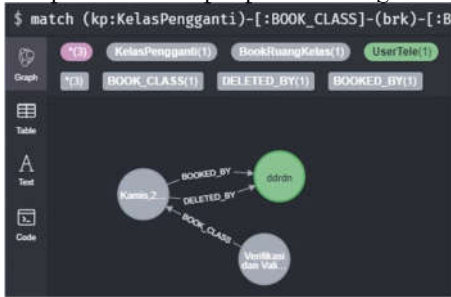
c. Tampilan hapus pesanan ruang kelas



Gambar. 17 Hapus pesanan ruang kelas

Gambar 17 adalah tampilan dari web *control chatbot*. Ruang kelas yang sudah terhapus atau kadaluwarsa bisa dilihat melalui website tersebut. *User* bisa menghapus secara manual jika ada kekeliruan.

d. Tampilan *record* hapus pesanan ruang kelas



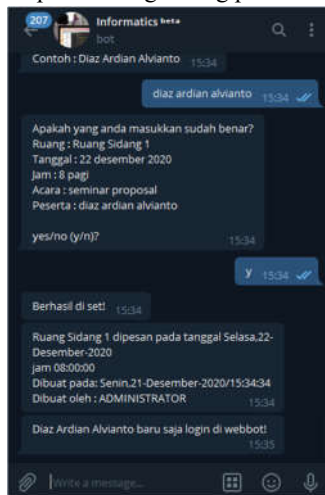
Gambar. 18 Tampilan *record* database hapus ruang kelas

Gambar 18 merupakan tampilan dari *record* hapus pesan kelas pengganti pada Neo4j. Pada gambar tersebut, ruang kelas yang dipesan oleh *user* sudah terhapus, maka *record* data pesanan ruang kelas yang tersimpan pada *database* akan diubah statusnya menjadi *history*.

4. Pesan ruang sidang.

Pada fitur ini, *user* dapat memesan ruang sidang untuk digunakan sebagai seminar sidang dan tidak dapat dipesan sebagai ruang perkuliahan. Alur pesan ruang sidang dalam penelitian ini antara lain:

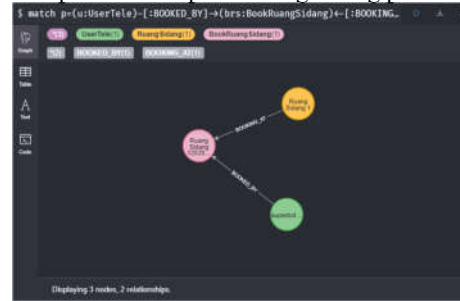
a. Tampilan pesan ruang sidang pada *chatbot*



Gambar. 19 Tampilan pesan ruang sidang

Gambar 19 merupakan tampilan dari halaman *chatbot* pada aplikasi telegram. Pada gambar tersebut, *user* memesan ruang sidang pada *chatbot*. Apabila *input* yang diberikan *user* dapat dimengerti, *bot* akan merespon pesan ruangan yang diminta oleh *user*. *Bot* akan memberikan respon ruangan sidang yang diinginkan berhasil dipesan oleh *user*.

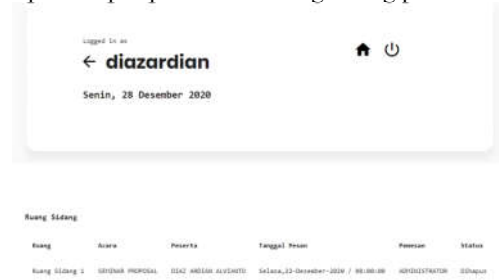
b. Tampilan *record* pesan ruang sidang pada neo4j



Gambar. 20 *Record* booking ruang sidang

Gambar 20 merupakan tampilan dari *record* pesan ruang sidang pada neo4j. Pada gambar tersebut, ruang sidang yang dipesan oleh *user* sudah tersimpan pada *database*.

c. Tampilan hapus pemesanan ruang sidang pada *chatbot*



Gambar. 21 Tampilan hapus pemesanan ruang sidang

Gambar 21 merupakan tampilan dari web *control chatbot* untuk ruang sidang yang sudah dipesan atau dihapus. Pada gambar tersebut, *user* web dapat menghapus ruang sidang pada web atau *chatbot* berdasarkan nama ruang sidang, tanggal, dan waktu pemesanan. Pada *chatbot*, apabila *input* yang diberikan *user* dapat dimengerti, *bot* akan merespon untuk menghapus pemesanan ruang sidang yang diminta oleh *user*. *Bot* akan memberikan respon ruangan sidang yang dipesan sebelumnya telah berhasil dihapus.

d. Tampilan *record* hapus pemesanan ruang sidang pada neo4j



Gambar. 22 Tampilan *record* hapus pemesanan ruang sidang

Gambar 22 merupakan tampilan dari *record* hapus pemesanan ruang sidang pada neo4j. Pada gambar tersebut,

ruang sidang yang dipesan oleh *user* sudah terhapus, maka *record* data pesanan ruang sidang yang tersimpan pada database akan hilang.

D. Hasil Performa Kecepatan Respon *Chatbot* Pada *Proxy Server* Ngrok.

Method	Status	Response Time
POST /webhook	200 OK	22.98ms
POST /webhook	200 OK	11.95ms
POST /webhook	200 OK	14.96ms
POST /webhook	200 OK	7.95ms
POST /webhook	200 OK	24.98ms
POST /webhook	200 OK	56.97ms
POST /webhook	200 OK	25.98ms
POST /webhook	200 OK	44.94ms
POST /webhook	200 OK	785.11ms
POST /webhook	200 OK	45.93ms
POST /webhook	200 OK	53ms
POST /webhook	200 OK	23.95ms
POST /webhook	200 OK	57.96ms

Gambar. 23 Hasil *request* ngrok

Ngrok dijalankan pada koneksi internet di Jurusan Teknik Informatika Unesa dengan rata rata kecepatan internet 30Mbps. Dari Gambar 23 dilakukan pengujian 20 *command* ke *chatbot* untuk melihat berapa kecepatan respon dari ngrok, dihasilkan waktu respon dengan rata rata 70ms pada jawaban text biasa dan 200 – 500ms pada jawaban data dari database.

IV. KESIMPULAN & SARAN

A. Kesimpulan

Perancangan Aplikasi *Chatbot* Rekomendasi Jadwal Penggunaan Ruang Berbasis *Graph Coloring* Menggunakan *Dialogflow* dan *Neo4j* yang dilakukan telah menghasilkan ketepatan hasil yang diminta untuk pesan ruang kelas dan ruang sidang dari *chatbot*, yaitu pada pembuatan jadwal menggunakan algoritma genetika untuk memperoleh jadwal yang dilakukan dengan data sampel pada semester gasal 2019/2020 menghasilkan tidak ada bentrok dari jadwal dosen dan kelas memperoleh *best fitness* dengan besar probabilitas *crossover* 0,75; probabilitas mutasi 0,50; dan kromosom/populasi 100. Sedangkan pada pengujian *graph coloring* menggunakan fitur K-1 *Coloring Algorithm* pada *Neo4j* telah mengasilkan rekomendasi penggunaan ruang kelas. Dengan percobaan 20 *keyword* dalam berbagai fitur yang telah disediakan seperti mencari kelas pengganti, pesan ruang kelas pengganti dan ruang sidang, serta hapus pemesanan ruang, mendapatkan hasil yang efektif. Dengan fitur ini, *bot* memberikan rekomendasi ruangan yang bisa digunakan sesuai dengan jumlah sks/ *timeslot*, pertimbangan jadwal dosen dan pertimbangan jadwal kelas. Rekomendasi ruang kelas juga

memungkinkan dari kelas yang sudah melaporkan bahwa matakuliah tidak diadakan pada waktu itu pada sistem *chatbot*. Dari penelitian yang telah dilakukan dapat disimpulkan bahwa ketepatan respon dari *chatbot* dapat diukur dari *keyword user* dengan parameter yang ada pada *Dialogflow*. *Dialogflow* mempelajari semua *keyword user says* yang sudah dimasukkan dalam *intens* dan akan di cek terdapat *action* pada *intens* yang terdapat *keyword user says*.

Penelitian ini juga telah melakukan pengujian terhadap performa kecepatan respon *chatbot* pada *Proxy Server* Ngrok, yaitu dengan melakukan 20 *keyword* ke *chatbot* untuk melihat berapa kecepatan respon dari *Ngrok*. Percobaan ini menghasilkan waktu respon dengan rata rata 70ms pada jawaban teks biasa dan 200 – 500ms pada jawaban data dari *database* yang dijalankan pada koneksi internet di Jurusan Teknik Informatika Unesa dengan rata rata kecepatan internet 30Mbps. Hal tersebut menunjukkan performa kecepatan respon *chatbot* bergantung pada koneksi internet dan juga jumlah *command* yang diberikan kepada *chatbot*.

Peneliti sangat berharap aplikasi *chatbot* rekomendasi jadwal penggunaan ruangan ini dapat membantu dalam menentukan jadwal mata kuliah dan ruang perkuliahan di Jurusan Teknik Informatika Universitas Negeri Surabaya. Sehingga perkuliahan dapat berjalan dengan baik sesuai dengan ruang yang tersedia dan diperoleh solusi optimal yang dapat pemeratakan penggunaan ruang dengan menyesuaikan kapasitas ruang dan jumlah peserta mata kuliah yang bersangkutan.

B. Saran

Untuk penelitian selanjutnya diharapkan aplikasi ini dapat dikembangkan lagi dengan metode atau algoritma yang baru dan data yang variatif agar menghasilkan sistem yang lebih baik lagi.

REFERENSI

- [1] K. R. Baker and D. Trietsch, "Principles of sequencing and scheduling: Second edition," *Princ. Seq. Sched.*, pp. 1–633, 2018.
- [2] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, M. Sterna, and J. Weglarz, *Handbook on Scheduling*. 2019.
- [3] R. Ganguli and S. Roy, "A Study on Course Timetable Scheduling using Graph Coloring Approach," *Int. J. Comput. Appl. Math.*, vol. 12, no. 2, pp. 469–485, 2017.
- [4] V. Nandhini, "A Study on Course Timetable Scheduling and Exam Timetable Scheduling using Graph Coloring Approach," *Int. J. Res. Appl. Sci. Eng. Technol.*, 2019.
- [5] A. Welch-powell and M. Eka, "Penerapan Teknik Pewarnaan Graph Pada Penjadwalan Ujian Dengan," vol. 6341, no. November, pp. 58–63, 2018.
- [6] S. Mirjalili, "Genetic algorithm," *Stud. Comput. Intell.*, vol. 780, pp. 43–55, 2019.
- [7] O. Kramer, *Studies in Computational Intelligence 679 Genetic Algorithm Essentials*. 2017.

- [8] Rosyid, "Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning - Algoritma Genetika," *Algoritm. Genet.*, vol. 91, pp. 399–404, 2017.
- [9] A. Singh, K. Ramasubramanian, S. Shivam, A. Singh, K. Ramasubramanian, and S. Shivam, "Introduction to Microsoft Bot, RASA, and Google Dialogflow," in *Building an Enterprise Chatbot*, 2019.
- [10] Neo4j, "Neo4j: The World Leading Graph Database," *Neo4J.Org*, 2016. .
- [11] V. First, G. Zhang, and C. Busch, "Chatbots: Development and Applications," 2017.
- [12] L. Yuan, L. Qin, X. Lin, L. Chang, and W. Zhang, "Effective and efficient dynamic graph coloring," in *Proceedings of the VLDB Endowment*, 2017.