

Penerapan Algoritma ACO untuk Penjadwalan Kuliah Pengganti pada Perguruan Tinggi

(Studi Kasus: Program Studi Informatika, Universitas Multimedia Nusantara)

Indah Noviasari¹, Andre Rusli², Seng Hansun³

Fakultas Teknik dan Informatika, Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang Indonesia

indahnovia3@gmail.com

andre.rusli@umn.ac.id

hansun@umn.ac.id

Diterima 12 November 2018

Disetujui 21 Desember 2018

Abstract— *Students and scheduling are both essential parts in a higher educational institution. However, after schedules are arranged and students has agreed to them, there are some occasions that can occur beyond the control of the university or lecturer which require the courses to be cancelled and arranged for replacement course schedules. At Universitas Multimedia Nusantara, an agreement between lecturers and students manually every time to establish a replacement course. The agreement consists of a replacement date and time that will be registered to the division of BAAK UMN which then enter the new schedule to the system. In this study, Ant Colony Optimization algorithm is implemented for scheduling replacement courses to make it easier and less time consuming. The Ant Colony Optimization (ACO) algorithm is chosen because it is proven to be effective when implemented to many scheduling problems. Result shows that ACO could enhance the scheduling system in Universitas Multimedia Nusantara, which specifically tested on the Department of Informatics replacement course scheduling system. Furthermore, the newly built system has also been tested by several lecturers of Informatics UMN with a good level of perceived usefulness and perceived ease of use.*

Keywords—*scheduling system, replacement course, Universitas Multimedia Nusantara, Ant Colony Optimization*

I. PENDAHULUAN

Perguruan Tinggi merupakan wahana tenaga ahli yang diharapkan mampu mengembangkan ilmu pengetahuan dan meningkatkan kualitas sumber daya manusia (SDM) [1]. SDM merupakan sumber daya yang terpenting yang dimiliki oleh suatu organisasi, dengan kata lain bahwa sumber daya manusia yang dapat membuat sumber-sumber lain dari sebuah organisasi dapat berfungsi dan mampu bergerak sehingga dapat mencapai tujuan yang telah ditetapkan [2]. Perguruan tinggi yang menjadi studi kasus pada penelitian ini adalah Universitas Multimedia Nusantara (UMN), khususnya program studi Informatika.

Badan Pusat Statistik Indonesia pada 2015 menuliskan bahwa setidaknya sudah ada kurang lebih 613.665 orang pada tahun 2014 yang meneruskan ke jenjang perguruan tinggi dari keseluruhan jumlah penduduk di Indonesia. Pada tahun 2015 angka tersebut naik mencapai satu juta orang lebih tinggi 60% dari tahun sebelumnya. Apabila hal tersebut berlangsung secara konsisten setiap tahunnya, maka Badan Pusat Statistik Indonesia memperkirakan pada tahun 2020 jumlah mahasiswa di Indonesia akan mencapai angka dua juta orang [3].

Selain mahasiswa, dalam suatu perguruan tinggi jadwal pada sebuah kegiatan belajar mengajar juga menjadi salah satu faktor yang penting, agar kegiatan tersebut berjalan dengan lancar tanpa adanya gangguan seperti bentrohnya jadwal. Penyusunan jadwal memiliki tantangan tersendiri karena penjadwalan harus dibuat sedemikian rupa sehingga tidak ada kegiatan pada waktu yang bersamaan [4]. Namun ketika penjadwalan jumlah mata kuliah, jumlah mahasiswa, jumlah ruangan, jumlah dosen serta waktu perkuliahan telah ditetapkan, terkadang dosen yang bersangkutan, justru berhalangan hadir, libur nasional, atau kejadian tidak terduga di luar kendali universitas atau dosen bersangkutan yang mengharuskan perkuliahan ditiadakan dan harus diatur kembali jadwal kuliah pengganti. Jadi, kuliah pengganti akan diadakan ketika hal-hal tersebut terjadi [5].

Penjadwalan kuliah pengganti diartikan bahwa sebelumnya sudah ada jadwal pasti pada mahasiswa dan dosen sehingga harus diatur kembali jadwal pengganti untuk menggantikan jadwal kuliah yang ditiadakan karena hal-hal penyebab kuliah ditiadakan [5]. Menurut Hery dari divisi BAAK UMN [6], prosedur penjadwalan kuliah pengganti harus melalui kesepakatan antara dosen dan mahasiswa yang bersangkutan. Kesepakatan tersebut berupa tanggal dan jam kuliah pengganti yang telah disepakati. Biro Administrasi Akademik dan Kemahasiswaan (BAAK) hanya bertugas dalam administrasi jadwal kuliah pengganti yang didaftarkan oleh dosen atau perwakilan kelas tersebut dan kemudian mencari

ruangan untuk jadwal kuliah pengganti tersebut serta memasukkan ke sistem yang nantinya jadwal kuliah pengganti yang telah didaftarkan akan ditampilkan di jadwal masing-masing mahasiswa dan dosen dalam kelas tersebut [6].

Menurut Hayi dan Ginting [7], pengaturan penjadwalan kuliah pengganti dapat dilakukan secara manual maupun secara otomatis (menggunakan *software*) yang mana dalam menentukan jadwal-jadwal tersebut membutuhkan waktu yang cukup lama dan cukup rumit. Dengan demikian, tidak jarang terjadi jadwal yang bentrok satu sama lain baik untuk mahasiswa, dosen ataupun ruangan kelas. Untuk memudahkan pengaturan penjadwalan kuliah yang efektif dan optimal melalui suatu proses otomatisasi, algoritma Ant Colony Optimization diimplementasikan agar mahasiswa dan dosen tidak mengalami jadwal yang bentrok satu sama lain [7].

Menurut Berlianty dan Arifin, algoritma Ant Colony Optimization (ACO) pertama kali dikemukakan oleh Marco Dorigo dan kawan-kawan pada tesis PhD tahun 1992 yang merupakan sebuah pendekatan awal terhadap berbagai masalah sulit seperti masalah Traveling Salesman Problem dan masalah tugas ganda (Quadratic Assignment Problem) [8]. Algoritma Ant Colony Optimization meniru tingkah laku semut ketika semut berada di dalam sebuah koloni untuk mencari sebuah sumber makanan. Algoritma semut lebih menitikberatkan pada perilaku kebiasaan atau yang sering dilakukan (dilewati) oleh semut. Semut memiliki keunikan dan kemampuan secara alami untuk menemukan alur atau lintasan terpendek dari sarangnya ke suatu sumber makanan tanpa pengertian visual atau penglihatan [9].

Menurut Lorena dan Akbar [10], algoritma Ant Colony Optimization dapat melakukan pengaturan penjadwalan kuliah secara otomatis dan cepat yang dibuat berdasarkan kesediaan waktu dari dosen dan kesediaan ruangan. Di sisi lain, penjadwalan dengan algoritma Ant Colony Optimization berhasil mengatasi bentrokan antar jadwal mahasiswa. Berdasarkan latar belakang permasalahan yang telah dijabarkan, penelitian ini bertujuan untuk mengimplementasikan dan mengetahui pengaruh penggunaan algoritma Ant Colony Optimization untuk penjadwalan kuliah pengganti pada perguruan tinggi, yang lalu pengujian dilakukan untuk mata kuliah yang terdapat pada program studi Informatika di Universitas Multimedia Nusantara.

II. LANDASAN TEORI

A. *Technology Acceptance Model*

Menurut Simarmata [11], *Technology Acceptance Model* (TAM) adalah teori yang mampu menjelaskan perilaku pengguna terhadap teknologi, penerimaan teknologi disebabkan oleh faktor *perceived ease of use*, *perceived usefulness*, dan pada akhirnya *actual use*, yang diusulkan oleh Davis pada tahun 1986.

Perceived usefulness didefinisikan sebagai sejauh mana seseorang percaya bahwa dengan menggunakan suatu teknologi akan meningkatkan performanya. *Perceived ease of use* didefinisikan sebagai sejauh mana seseorang percaya bahwa menggunakan suatu teknologi akan terbebas dari usaha fisik maupun mental [11]. Pertanyaan untuk *perceived usefulness* mengacu pada pertanyaan yang meyakinkan seseorang apakah teknologi yang digunakan akan memberikan manfaat dan pertanyaan untuk *perceived ease of use* mengacu pada pertanyaan yang meyakinkan seseorang apakah teknologi tersebut mudah digunakan [12].

B. *Penjadwalan & Kuliah Pengganti pada Perguruan Tinggi*

Di perguruan tinggi, program penjadwalan merupakan salah satu hal penting dalam proses belajar mengajar, karena semua kegiatan dosen dan mahasiswa bergantung pada jadwal yang ada, sehingga harus disusun optimasi penjadwalan sehingga tidak ada jadwal dalam waktu yang bersamaan antara dosen dan mahasiswa [13]. Dikarenakan dalam penyusunan jadwal memiliki tantangan tersendiri yang cukup rumit agar tidak ada kegiatan pada waktu yang bersamaan [4].

Kuliah pengganti diartikan bahwa sebelumnya sudah ada jadwal pasti pada mahasiswa dan dosen sehingga harus diatur kembali jadwal pengganti untuk menggantikan jadwal kuliah yang ditiadakan [5]. Dosen yang berhalangan hadir karena alasan yang dibenarkan atau alasan tertentu menurut peraturan diharuskan melapor kepada penanggung jawab mata kuliah dan menentukan solusi yaitu dosen yang berhalangan hadir memberikan jadwal kuliah pengganti pada waktu lain yang telah disepakati bersama oleh dosen dan mahasiswa [14].

C. *Algoritma Ant Colony Optimization*

Algoritma Ant Colony Optimization (ACO) merupakan teknik probabilistik untuk menjawab masalah komputasi yang bisa dikurangi dengan menemukan jalur yang baik dengan graf. ACO pertama kali dikembangkan oleh Marco Dorigo pada tahun 1992. Sesuai dengan nama algoritmanya, ACO diinspirasi oleh koloni semut karena tingkah laku semut yang menarik ketika mencari makanan. Semut-semut menemukan jarak terpendek antara sarang semut dan sumber makanannya. Ketika berjalan dari sumber makanan menuju sarang semut, semut memberikan tanda dengan zat feromon sehingga akan tercipta jalur feromon. Feromon adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis. Berbeda dengan hormon, feromon menyebar ke luar tubuh dan hanya dapat dikenali oleh individu lain yang sejenis. Semut dapat mencium feromon dan ketika semut memilih jalur semut, semut cenderung memilih jalur yang ditandai oleh feromon dengan konsentrasi yang tinggi.

Apabila semut telah menemukan jalur yang terpendek maka semut-semut akan terus melalui jalur tersebut. Jalur lain yang ditandai oleh feromon lama akan memudar atau menguap, seiring berjalannya waktu. Jalur-jalur yang pendek akan mempunyai ketebalan feromon dengan probabilitas yang tinggi dan membuat jalur tersebut akan dipilih dan jalur yang panjang akan ditinggalkan.

Jalur feromon membuat semut dapat menemukan jalan kembali ke sumber makanan atau sarang semut. Algoritma ACO telah banyak digunakan untuk menghasilkan penyelesaian yang mendekati optimal [15]. Aplikasi algoritma semut dalam kehidupan sehari-hari mencakup beberapa persoalan sebagai berikut [16].

- a. Traveling Salesman Problem (TSP), yaitu mencari jalur terpendek dalam sebuah graf menggunakan jalur Hamilton.
- b. Quadratic Assignment Problem (QAP) yang berusaha menempatkan sejumlah sumber n pada sejumlah m lokasi dengan meminimalkan biaya assignment.
- c. Job-shop Scheduling Problem (JSP), juga salah satu contoh aplikasi algoritma semut untuk menjadwalkan sejumlah j pekerjaan menggunakan sejumlah m mesin sehingga seluruh pekerjaan diselesaikan dalam waktu yang minimal.

Pewarnaan graf koloni semut yang nyata dan *artificial* terdapat banyak kemiripan [8]. Keduanya terbentuk dari sebuah populasi yang terdiri dari individu-individu yang berkerja sama untuk mencapai tujuan. Semut *artificial* hidup di dunia virtual, karenanya semut hanya memodifikasi nilai numerik (disebut analogi *artificial* pheromones) yang berhubungan dengan keadaan-keadaan permasalahan yang berbeda. Sebuah rangkaian dari nilai-nilai feromon yang berhubungan dengan keadaan permasalahan disebut pheromone trail atau jejak feromon. Mekanisme untuk evaporation atau penguapan feromon pada koloni semut nyata yang membuat semut *artificial* dapat melupakan sejarah (jalur-jalur yang pernah diambil) dan fokus pada arah pencarian baru yang menjanjikan. Seperti semut-semut nyata, semut-semut *artificial* membuat solusi secara berurut dengan bergerak dari satu keadaan permasalahan ke lainnya. Semut-semut nyata hanya berjalan, memilih arah berdasarkan konsentrasi feromon lokal dan kebijakan keputusan stokastik. Semut *artificial* membuat solusi sedikit demi sedikit, dan bergerak dari keadaan permasalahan yang tersedia dan membuat keputusan stokastik setiap langkah. Meskipun begitu, terdapat perbedaan antara yang nyata dan semut *artificial* sebagai berikut.

- a. Semut *artificial* hidup di dunia dan pada waktu diskrit, semut berpindah secara sekuen melewati setiap batasan dari permasalahan.
- b. Update feromon (penumpukan dan penguapan feromon) tidak dilakukan dengan

jalan yang sama pada semut yang nyata dan semut *artificial*. Update feromon dilakukan oleh beberapa dari semut *artificial* dan terkadang dilakukan saat solusi telah dibangun.

- c. Beberapa implementasi dari semut *artificial* menggunakan mekanisme tambahan yang tidak ada pada semut-semut nyata, seperti local search, backtracking, dan lain-lain.

Dalam algoritma semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek [10], yaitu:

Langkah 1 :

- a. Inisialisasi harga parameter-parameter algoritma.

Parameter-parameter yang diinisialisasikan adalah.

- Intensitas jejak semut antar titik dan perubahannya (τ_{ij}).
- Banyak titik.
- titik berangkat dan titik tujuan.
- Banyak semut.
- Tetapan penguapan jejak semut (ρ), nilai ρ dimulai dari 0 sampai dengan 1 untuk mencegah jejak feromon yang tak terhingga.
- Jumlah siklus maksimum (NCmax) bersifat tetap selama algoritma dijalankan, sedangkan τ_{ij} akan selalu diperbaharui nilainya pada setiap siklus algoritma mulai dari siklus pertama (NC=1) sampai tercapai jumlah siklus maksimum (NC=NCmax) atau sampai terjadi konvergensi.

- b. Inisialisasi titik pertama setiap semut.

Setelah inisialisasi pada poin pertama dilakukan, kemudian m semut ditempatkan pada titik pertama.

Langkah 2 :

Pengisian titik pertama ke dalam tabu list. Hasil inisialisasi titik pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen pertama tabu list. Hasil dari langkah ini adalah terisinya elemen pertama tabu list setiap semut dengan indeks titik tertentu, yang berarti bahwa setiap $[\text{tabu}]_k$ (1) bisa berisi indeks titik antara 1 sampai n sebagaimana hasil inisialisasi pada langkah 1.

Langkah 3 :

Penyusunan rute kunjungan setiap semut ke setiap titik. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap titik, akan mulai melakukan perjalanan dari titik pertama, masing-masing sebagai titik asal dan salah satu titik lainnya sebagai titik tujuan, kemudian dari titik kedua masing-masing,

koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari wilayah yang tidak terdapat pada [tabu]-k sebagai titik tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua titik satu persatu dikunjungi atau telah menemptati [tabu]-k.

Langkah 4 :

- a. Memperbarui intensitas jejak kaki semut antar titik.

Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar titik yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan nilai intensitas jejak kaki semut antar titik. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antar titik untuk siklus selanjutnya dihitung dengan persamaan (2.1).

$$\tau_{ij} = \rho + \tau_{ij}$$

Dimana :

τ_{ij} = Intensitas jejak semut antar titik

ρ = Tetapan penguapan jejak semut

Langkah 5 :

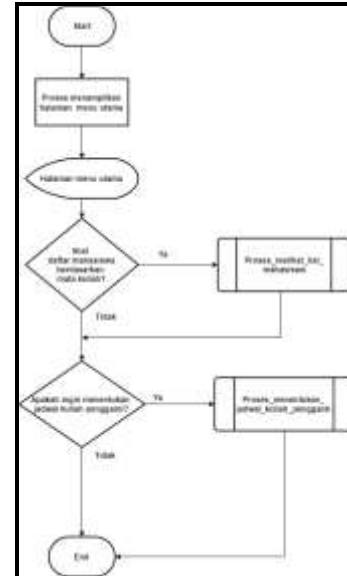
Pengosongan tabu list, dan ulangi langkah 2 jika diperlukan. Tabu list perlu dikosongkan untuk diisi lagi dengan urutan titik yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi, algoritma diulang lagi dari langkah 2 dengan harga parameter intensitas jejak kaki semut antar titik yang sudah diperbaharui.

III. METODE DAN PERANCANGAN SISTEM

A. Metodologi Penelitian

Metodologi penelitian yang digunakan pada penelitian ini dapat dipecah menjadi beberapa tahapan utama, yaitu Telaah Literatur, Perancangan Sistem, Pemrograman Sistem, Uji Coba dan Evaluasi, dan Penulisan Laporan.

B. Perancangan Aplikasi

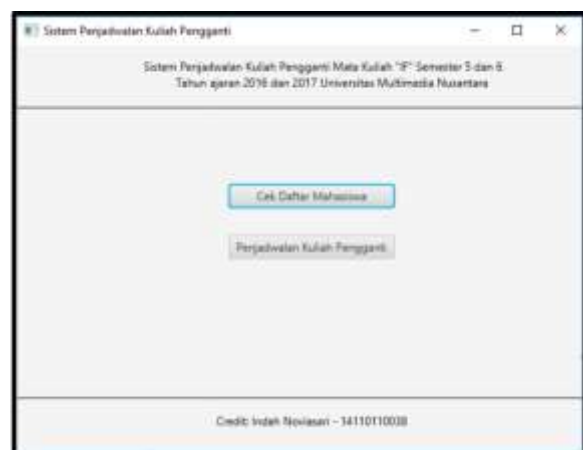


Gambar 1. Flowchart Halaman Menu Utama

Aplikasi yang dibangun merupakan aplikasi berbasis desktop. Perancangan yang dilakukan adalah dengan membuat flowchart dari aplikasi. Berikut adalah flowchart utama dari sistem. Gambar 1 merupakan flowchart untuk Halaman Menu Utama. Pada halaman ini dilakukan proses menampilkan tampilan halaman Menu Utama yang berisi dua pilihan yaitu user dapat memilih untuk melihat daftar mahasiswa berdasarkan mata kuliah dan untuk menentukan jadwal kuliah pengganti.

IV. IMPLEMENTASI DAN UJI COBA

Pada bagian ini dijelaskan mengenai hasil implementasi dari desain user interface yang telah dilakukan dalam bentuk screenshot aplikasi. Dalam proses implementasi, pembangunan aplikasi menggunakan bahasa pemrograman Java.



Gambar 2. Halaman Menu Utama

Gambar 2 merupakan implementasi dari halaman menu utama pada sistem. Pada halaman menu utama terdapat dua pilihan untuk user yaitu untuk melihat

daftar mahasiswa dan untuk menentukan jadwal kuliah pengganti.

Gambar 3. Halaman Cek Daftar Mahasiswa

Gambar 3 merupakan implementasi dari halaman cek daftar mahasiswa. Halaman ini akan menampilkan sebuah form yang memiliki field Mata Kuliah dan Kelas yang perlu dimasukkan oleh user untuk dapat melihat daftar mahasiswa berdasarkan mata kuliah. Hasil akan menampilkan nama mahasiswa, nomor induk mahasiswa berdasarkan mata kuliah dan kelas yang ditentukan. Terdapat tombol 'Cari' yang digunakan untuk mencari dan tombol 'Kembali ke menu utama' untuk kembali ke halaman menu utama.

Nama Mahasiswa	Nomor Induk Mahasiswa
EDWIN HANDOKO	14110110007
STEVEN SUJIRTA	14110110014
KEVIN LIONERY	14110110020
SINTYA OKTAVIANI	14110110021
SASA INYASA	14110110031
ADRIAN HARTANTO	14110110033
PHERRIS	14110110034
INDAH KURNATI	14110110035
MELDA LENA	14110110036
JEREMY EWAR HALIM	14110110057
TOMMY ADUSTIAN	14110110069
THOMAS ARIEL	14110110073
VINCENCUS IVAN PRATAMA	14110110089

Gambar 4. Halaman Hasil Pencarian Mahasiswa

Gambar 4 merupakan implementasi dari halaman hasil pencarian mahasiswa untuk menampilkan daftar mahasiswa berdasarkan mata kuliah dan tombol 'Kembali' untuk kembali ke halaman cek daftar mahasiswa.

Tanggal	Jam Mulai	Jam Akhir	Aksi
2018-06-12	08:00	11:00	Hapus
2018-06-13	08:00	11:00	Hapus
2018-06-20	14:00	17:00	Hapus

Gambar 5. Halaman Penjadwalan Kuliah Pengganti

Gambar 5 merupakan implementasi dari halaman penjadwalan kuliah pengganti. Halaman akan menampilkan sebuah form yang memiliki field yang perlu dimasukkan oleh user untuk mendapatkan jadwal kuliah pengganti yang sesuai. Terdapat tombol 'Tambah' untuk menambahkan jadwal kuliah pengganti yang ditentukan dan akan ditampung ke dalam tabel pilihan jadwal kuliah pengganti. Tombol 'Hapus' pada tabel yang menampilkan jadwal kuliah pengganti yang dipilih untuk menghapus jadwal kuliah pengganti yang ditambahkan sebelumnya. Tombol 'Hitung Penjadwalan' untuk menghitung dan mencari hasil penjadwalan kuliah pengganti yang sesuai. Tombol 'Kembali ke menu utama' untuk kembali ke halaman menu utama.

Tanggal	Jam Mulai	Jam Akhir	Jumlah Mahasiswa Beresita
2018-06-20	14:00	17:00	29
2018-06-13	08:00	11:00	29
2018-06-12	08:00	11:00	29

Gambar 6. Halaman Hasil Penjadwalan Kuliah Pengganti

Nama Mahasiswa	Nomor Induk Mahasiswa
EDWIN HARDOKO	54110110007
TEJEN SUHIRA	54110110014
KEVIN LONDER	54110110030
UNTYA OKTAVIANI	54110110021
RAISA KHAGA	54110110031
ADRIAN HARTANTO	54110110033
THERIS	54110110034
INDAH KURNIATI	54110110035
MELDA LENA	54110110036
ERENY EVAN HALBI	54110110037
TOMMY AGUSTIAN	54110110069
THOMAS ARELL	54110110073
PRINCESIA IVAN PRATAMA	54110110088

Gambar 7. Halaman Jumlah Mahasiswa Tersedia

Gambar 6 merupakan implementasi dari halaman hasil penjadwalan kuliah pengganti untuk menampilkan hasil penjadwalan kuliah pengganti dan tombol 'Kembali' untuk kembali ke halaman penjadwalan kuliah pengganti. Jika user klik sebanyak dua kali pada masing-masing kolom Jumlah Mahasiswa Bersedia maka akan ditampilkan mahasiswa yang dapat menghadiri kuliah pengganti pada jam dan tanggal tersebut seperti pada Gambar 7.

Setelah sistem selesai dirancang dan dibangun dilakukan pengujian yang dibagi menjadi tiga pengujian. Pada pengujian bagian pertama bertujuan untuk mengetahui bahwa jumlah mahasiswa yang dapat hadir pada kuliah pengganti dalam proses otomatisasi sistem sama dengan jumlah mahasiswa yang dicek secara manual berdasarkan jadwal reguler masing-masing mahasiswa. Pada pengujian bagian kedua dilakukan dengan simulasi yang bertujuan untuk mengetahui tanggal dan jam kuliah pengganti yang paling banyak memiliki jumlah mahasiswa yang dapat hadir pada kuliah pengganti tersebut. Pada pengujian ketiga bertujuan untuk mendapatkan feedback dari responden tentang kegunaan sistem untuk memperkuat hasil penelitian

Pengujian pertama menggunakan data kuliah pengganti yang telah terjadi pada 10 kelas di semester 5 dan 6 tahun ajaran 2016-2017 program studi Informatika UMN dan didapatkan hasil bahwa jumlah mahasiswa yang dapat hadir pada kuliah pengganti dalam proses otomatisasi sistem 100% sama dengan jumlah mahasiswa yang dicek secara manual berdasarkan jadwal reguler masing-masing mahasiswa. Pengujian pertama menunjukkan bahwa hasil output pada sistem benar dan sesuai.

Pengujian kedua dilakukan dengan simulasi yang menggunakan tanggal dan jam kuliah pengganti baru pada 2 kelas di semester 5 dan 6 tahun ajaran 2016-2017 program studi Informatika UMN dan didapatkan hasil bahwa sistem dapat menentukan bahwa dari beberapa banyak pilihan jadwal kuliah pengganti, sistem dapat menentukan salah satu dari beberapa pilihan tanggal dan jam tersebut dengan mahasiswa yang paling banyak dapat menghadiri. Hal ini membuktikan bahwa mahasiswa yang berperan

sebagai semut pada algoritma ACO akan memilih tanggal dan jam kuliah pengganti yang berperan sebagai jalur dengan jejak feromon yang paling banyak karena pada jalur tersebut mahasiswa tidak mengalami bentrokan jadwal sehingga jejak feromon pada jalur tersebut akan bertambah dan mahasiswa (semut) akan memilih jalur tersebut

Pengujian ketiga menggunakan dua responden yang merupakan dosen program studi Informatika Universitas Multimedia Nusantara. Pengujian dilakukan dengan cara studi lapangan dimana responden diminta untuk menggunakan sistem penjadwalan kuliah pengganti dan setelah itu dilakukan wawancara kepada responden. Dari wawancara tersebut berisi 5 pertanyaan berdasarkan *perceived ease of use* dan *perceived usefulness* dari Technology Acceptance Model (TAM) dan didapatkan hasil bahwa responden pertama setuju bahwa sistem penjadwalan kuliah pengganti sangat berguna dan memudahkan responden sebagai dosen untuk menentukan kuliah pengganti, setelah itu responden pertama juga berpendapat bahwa fitur pada sistem sangat jelas ketika digunakan serta menurut responden pertama, sistem penjadwalan kuliah pengganti ini membuat penjadwalan kuliah pengganti menjadi cepat dan tidak membutuhkan waktu yang lama sehingga membuat penjadwalan kuliah pengganti menjadi efisien. Dan responden kedua setuju bahwa sistem penjadwalan kuliah pengganti sangat berguna dan sangat memudahkan responden sebagai dosen untuk menentukan kuliah pengganti, setelah itu responden kedua juga berpendapat bahwa fitur mudah dipahami ketika digunakan dan membuat penjadwalan kuliah pengganti menjadi lebih cepat.

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan penelitian yang dilakukan, disimpulkan bahwa Algoritma Ant Colony Optimization untuk menentukan jadwal kuliah pengganti pada perguruan tinggi, dengan studi kasus pada program studi Informatika, Universitas Multimedia Nusantara, telah selesai dilaksanakan dan hasil pengujian menunjukkan bahwa jumlah mahasiswa yang dapat hadir pada kuliah pengganti dalam proses otomatisasi sistem 100% sama dengan jumlah mahasiswa yang dicek secara manual berdasarkan jadwal reguler masing-masing mahasiswa dan dapat menentukan bahwa dari beberapa banyak pilihan jadwal kuliah pengganti, sistem dapat menentukan salah satu dari beberapa pilihan tanggal dan jam tersebut dengan mahasiswa yang paling banyak dapat menghadiri.

Selain itu, sistem penjadwalan kuliah pengganti telah diterima dari aspek *perceived usefulness* dan *perceived ease of use* yang didasari oleh TAM. Hal ini dibuktikan dengan hasil wawancara dengan dua responden sebagai dosen yang menyatakan bahwa sistem penjadwalan kuliah pengganti sangat berguna dan memudahkan pengguna untuk menentukan kuliah

pengganti yang mengacu pada aspek *perceived usefulness*, serta sistem sangat jelas saat digunakan dan tidak membutuhkan waktu yang lama dalam menentukan kuliah pengganti yang mengacu pada aspek *perceived ease of use*.

B. Saran

Berikut ini merupakan beberapa saran dan pengembangan selanjutnya dari hasil penelitian yang telah didapat.

1. Integrasi secara langsung dengan database pusat UMN, sehingga semua mata kuliah ada pada sistem ini, data jadwal perkuliahan pada sistem tersinkronisasi secara real time dan dapat digunakan oleh semua dosen Universitas Multimedia Nusantara.
2. Untuk pengembangan lebih lanjut, ketika sudah terintegrasi secara langsung dengan database pusat UMN dapat menambahkan fitur Login agar dosen hanya dapat memilih kelas yang sedang diajar pada semester dan tahun ajaran tersebut.

DAFTAR PUSTAKA

- [1] Artawan, I. M. (2002). Strategi Meningkatkan Mutu Pendidikan di Perguruan Tinggi.
- [2] Jalaludin. (2012). Latar Belakang Penelitian: Sumber Daya Manusia. Efektivitas Manajemen Pengembangan Kinerja Dosen UPI.
- [3] Statistik, B. P. (2017, May 2014). Jumlah Perguruan Tinggi, Mahasiswa, dan Tenaga Edukatif (Negeri dan Swasta) Menurut Provinsi 2013/2014 dan 2014/2015. [Online] dari Badan Pusat Statistik: <https://www.bps.go.id/linkTabelStatis/view/id/184>. Diakses pada 09 September 2017.
- [4] Fernandez, A., Handoyo, E., & Somantri, M. (2009). Pembangunan Aplikasi Penyusun Jadwal Kuliah Menggunakan Algoritma Semut.
- [5] Winatha, D. N. (2010). Program Bantu Pemilihan Jadwal Kuliah Pengganti Dan Tambahan Dengan Menggunakan Teori Himpunan.
- [6] Hery. (2017). Prosedur Penjadwalan Kuliah Pengganti Di Universitas Multimedia Nusantara. (I. Noviasari, Pewawancara).
- [7] Ginting, S. B., & Akbar, H. (2011). Pembangunan Perangkat Lunak Menggunakan Algoritma Ant Colony Optimization untuk Optimalisasi Penjadwalan Kuliah (Studi Kasus Penjadwalan Kuliah Jurusan Teknik Komputer UNIKOM).
- [8] Dorigo, M., Birattari, M., & Stützle, T. (2006). Artificial Ants as a Computational Intelligence Technique. *IEEE Computational Intelligence Magazine*, 28-39.
- [9] Karjono, Budiyo, U., & Moedjiono. (2016). Aplikasi Penjadwalan Crew Ship Menggunakan Ant Colony Optimization: Studi Kasus PT Scorpa Pranedya.
- [10] Lorena, S., & Akbar, H. (2011). Pembangunan Perangkat Lunak Menggunakan Algoritma Ant Colony Optimization Untuk Optimalisasi Penjadwalan Kuliah. 2-16.
- [11] Simarmata, M. (2015). Model Penerimaan Teknologi (Technology Acceptance Model). Medan: Universitas HKBP Nommensen.
- [12] Rahmah, N. (2017, October). Technology Acceptance Model (TAM). [Online] dari Dictio Website: www.dictio.id. Diakses pada 25 Juli 2018.
- [13] Ariani, D., Fahriza, A., & Prasetyaningrum, I. (2012). Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika PENS Dengan Menggunakan Algoritma PSO.
- [14] TIM SPMI UNDIP. (2011). Perkuliahan Tamu Area Bidang Akademik. Semarang.
- [15] Bullnheimer, B., Hartl, R. F., & Strauss, C. (1997). An improved ant system algorithm for the vehicle. 1.
- [16] Leksono, A. (2009). Algoritma Ant Colony Optimization Untuk Menyelesaikan Traveling Salesman Problem.

