

Membangun Sistem Penjadwalan Ruang Laboratorium dengan Algoritma *Modified BiDirectional A**

M.Ridwan¹, Elvia Budianita²

Teknik Informatika UIN Suska Riau

Jl.H.R Subrantas No.155 Simpang Baru Panam Pekanbaru, Telp.0761-56223i

e-mail: ridwan.spybeam@gmail.com, elvia.budianita@gmail.com

Abstrak

Sistem penjadwalan ruang laboratorium merupakan sistem terkomputerisasi yang berfungsi untuk menyusun jadwal kelas-kelas praktikum yang akan menggunakan laboratorium. Pada saat ini, penyusunan jadwal laboratorium jurusan Teknik Informatika UIN Suska masih dilakukan secara manual sehingga kepala laboratorium harus mengumpulkan sendiri data yang dibutuhkan dalam penyusunan jadwal. Teknik analisis data pada sistem ini menggunakan metode pembangunan perangkat lunak secara waterfall. Proses penyusunan jadwal dilakukan dengan metode MBDA (Modified Bidirectional A*) dengan penentuan bobot berdasarkan kategori sisa waktu terbuang, kelas yang berulang, dan status dosen yang telah terjadwal. Pada MBDA* metode pencarian heuristik dilakukan dan setiap kandidat solusi akan disimpan kedalam struktur data graph yang memiliki bobot. Algoritma MBDA akan menelusuri simpul tersebut dan mencari solusi terbaik berdasarkan total bobot terendah. Berdasarkan pengujian terhadap 10 kasus secara acak, seluruh kasus menghasilkan jadwal yang bebas bentrokan waktu pengajar ataupun mahasiswa dan sesuai dengan waktu kosong yang diajukan(100%).

Kata kunci: Penjadwalan, pencarian heuristic, euclidean distance, MBDA.

1. Pendahuluan

Penjadwalan merupakan suatu proses yang sangat penting untuk mengatur rangkaian kegiatan agar dapat berjalan dengan teratur dan terhindar dari bentrokan. Terbukti dengan banyaknya penelitian terkait penjadwalan dengan berbagai metode dalam penyelesaiannya. Diantaranya adalah, Aplikasi Pewarnaan *Graph* terhadap Penjadwalan Kuliah di Jurusan Matematika UIN Malang [1], Design dan Implementasi Jadwal Kuliah dengan Menggunakan Algoritma Semut Berbasis Web [2], Sistem Penjadwalan Dokter Jaga dengan Menggunakan Algoritma Greedy dengan Permutasi [3], Membangun Aplikasi Penjadwalan Asisten Praktikum di STMIK AMIKOM Yogyakarta [4], dan kasus-kasus penjadwalan yang lainnya.

Kepala laboratorium akan menyesuaikan waktu yang diajukan seluruh dosen yang akan menggunakan laboratorium terhadap ruangan laboratorium yang tersedia. Hal tersebut tentunya akan sulit dilakukan ketika variable yang disusun dalam penjadwalan berjumlah sangat banyak, sementara penyusunan jadwal tersebut dilakukan secara manual. Untuk itu, dirancanglah sistem penjadwalan terkomputerisasi dan metode *Modified BiDirectional A**(MBDA) akan diterapkan pada proses pencarian solusinya.

Algoritma MBDA* adalah salah satu jenis dari pengembangan algoritma A* yang menggunakan metode penghitungan bobot dalam penelusuran *graph* untuk menemukan solusinya. Pada penerapannya, algoritma ini dapat diterapkan pada kasus-kasus yang di representasikan melalui graph berbobot. Dalam pencarian solusinya, kasus penjadwalan juga dapat direpresentasikan dalam bentuk graph berbobot.

Berdasarkan eksperimen yang dilakukan oleh Tetsuo Shibuya terhadap jaringan jalan di Tokyo membuktikan bahwa jumlah simpul yang dibangkitkan MBDA* adalah setengah dari jumlah simpul yang dibangkitkan oleh A*[5]. Cara kerja algoritma ini yang melakukan pencarian solusi dengan dua arah, dapat mempersingkat waktu pencarian solusi karena pencarian dari *node* awal ke *node* goal dan sebaliknya dilakukan secara serentak, membuat jumlah *node* yang di bangkitkan lebih sedikit dibandingkan algoritma A*. Pada penelitian ini, akan dibahas mengenai penerapan algoritma MBDA* pada kasus penjadwalan ruang laboratorium yang telah direpresentasikan kedalam graf berbobot. Komponen-komponen yang disusun dalam penjadwalan terdiri dari waktu-waktu kosong yang diajukan oleh dosen/asisten, waktu kosong yang dimiliki mahasiswa setiap mata praktikum, kelas praktikum, serta ruangan laboratorium.

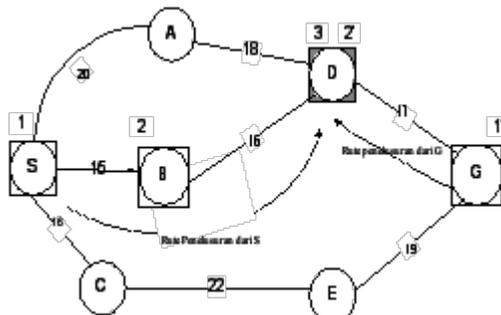
Setiap dosen memiliki prioritas yang sama dalam penyusunan jadwal, tanpa memperhatikan jabatan, usia dosen, pendidikan terakhir dosen ataupun kriteria dosen lainnya. Sistem tidak memberikan solusi untuk keterbatasan ruangan laboratorium serta tidak ada kelas praktikum yang harus diletakkan pada ruangan tertentu. Tujuan yang ingin dicapai dalam penelitian ini adalah untuk

merancang dan membangun sistem penjadwalan penggunaan laboratorium terkomputerisasi dan menerapkan algoritma MBDA* dalam proses pencarian solusinya.

2. Metode Penelitian

2.1. Metode MBDA

MBDA merupakan variasi pengembangan algoritma A*. Algoritma A* merupakan algoritma pencarian yang menggunakan fungsi *heuristic* untuk menuntun pencarian rute, khususnya dalam hal pengembangan dan pemeriksaan simpul-simpul pada graf [6]. Dengan menerapkan suatu fungsi *heuristic*, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan [5]. Algoritma A* melakukan penelusuran satu arah dari state awal ke state goal. Sedangkan MBDA melakukannya dengan dua arah secara serentak, yaitu dari state awal ke akhir dan dari state akhir ke state awal [7].



Gambar 1. Penelusuran solusi jalur terpendek dengan MBDA

Perbedaan lainnya adalah pada algoritma MBDA dilakukan modifikasi pada fungsi heuristik. Karena pencarian solusi dilakukan melalui dua arah, maka persamaan fungsi heuristik juga terdiri dari dua jenis, yaitu fungsi heuristik pencarian maju dan fungsi heuristik pencarian mundur. Sehingga fungsi heuristik untuk simpul n pada pencarian maju (dari simpul *start* ke simpul *goal*) adalah: [8]

$$f = g(S, n) + \frac{1}{2} [h_s(n) - h_g(n)] \quad (1)$$

Sedangkan fungsi heuristik untuk simpul n pada pencarian mundur (dari simpul *goal* ke simpul *start*) adalah:

$$f = g(G, n) + \frac{1}{2} [h_s(n) - h_g(n)] \quad (2)$$

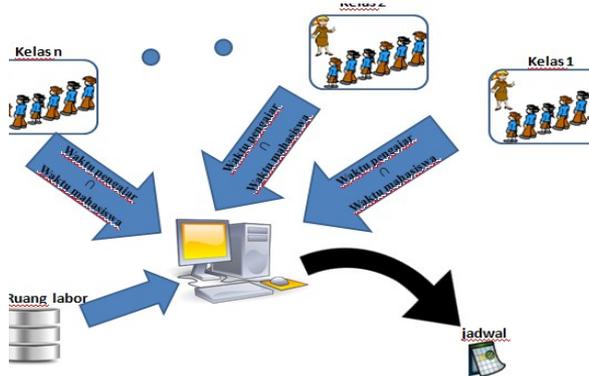
Dimana :

- $g(S, n)$ = Biaya sebenarnya dari S ke n
- $g(G, n)$ = Biaya sebenarnya dari G ke n
- $h_s(n)$ = Biaya perkiraan dari n ke G
- $h_g(n)$ = Biaya perkiraan dari n ke S

Pencarian pada MBDA akan berhenti di titik tengah dimana terjadi pertemuan solusi antara pencarian dari state awal dan pencarian dari state akhir.

2.2. Analisa Penerapan untuk Penjadwalan

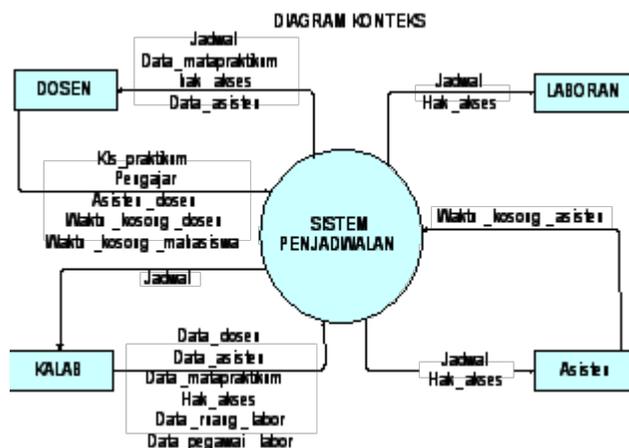
Secara umum, proses penjadwalan dapat digambarkan melalui gambar berikut.



Gambar 2. Penjadwalan penggunaan ruang laboratorium secara umum

Setiap kelas praktikum yang akan dijadwalkan untuk menggunakan ruangan laboratorium tentunya memiliki waktu kosong pengajar dan mahasiswa yang berbeda-beda. sistem ini akan mengelola waktu kosong yang dimiliki pengajar dan mahasiswa setiap kelas.

Pemodelan fungsional sistem meliputi *Data Flow Diagram* (DFD) yang terdiri dari *Context Diagram* (Diagram konteks), DFD level 1, dan DFD level 2. DFD merupakan suatu diagram yang menjelaskan alur data pada suatu sistem dengan menggunakan notasi-notasi tertentu, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas. Diagram konteks digambarkan pada gambar berikut:



Gambar 3. Diagram konteks sistem penjadwalan

Berikut adalah langkah-langkah yang dilakukan pada tahap penyusunan jadwal:

1. Mengumpulkan semua simpul kemungkinan jadwal

Pada tahap ini sistem akan mengumpulkan semua simpul sebagai kandidat solusi dalam penyusunan jadwal.



Gambar 4. Penentuan waktu pengajar dan mahasiswa secara bersamaan.

Dari semua waktu kosong yang dimiliki setiap pengajar ditentukan waktu yang dapat digunakan secara bersama oleh setiap pengajar.

Contoh Kelas A (3 sks) :

Mahasiswa =Senin(08:00-10:30),
Selasa(10:30-13:00),
Rabu(08:00-12:10)

Pengajar =Senin(08:00-12:10),
Selasa(10:30-14:40)

Dari contoh diatas dapat disimpulkan bahwa waktu mahasiswa dan pengajar terdiri dari Senin(08:00-10:30), dan Selasa(10:30-13:00). Sehingga didapat dua simpul kemungkinan jadwal atau kandidat solusi, yaitu KelasASenin08:00-10:30, KelasASelasa10:30-13:00.

2. Merepresentasikan kasus dalam bentuk *graph*

Setelah semua simpul terkumpul, setiap simpul akan menjadi *state* yang akan direpresentasikan dalam bentuk struktur data *graph*. *Graph* yang digunakan adalah *graph* berarah. Terdapat dua aturan dalam proses penyusunan simpul graf, yaitu:

- Simpul suksesor adalah waktu simpul yang dapat menjadi urutan yang sesuai bagi simpul parent.
- Simpul dapat menjadi suksesor apabila tidak ada simpul lain yang dapat mengisi ruang waktu diantaranya.

Jika terdapat data simpul sebagai berikut:

ASenin08:00-10:30,
BSelasa10:30-12:10,
CSenin08:50-11:20,
DSenin10:30-12:10.

Dari simpul diatas simpul A tidak dapat dijadikan *parent* bagi simpul C ($A \rightarrow C$), karena A bukan merupakan urutan waktu yang sesuai. Seharusnya urutan yang sesuai dengan A adalah simpul yang

memiliki waktu hari senin jam 10:30 keatas. Simpul D dapat dijadikan suksesor simpul A(A→D), karena keduanya merupakan urutan waktu yang sesuai. Simpul A dan B merupakan urutan waktu yang sesuai, namun keduanya tidak dapat dihubungkan, karena diantara simpul lain terdapat simpul yang lebih dekat jarak waktunya terhadap simpul A, yaitu simpul D.

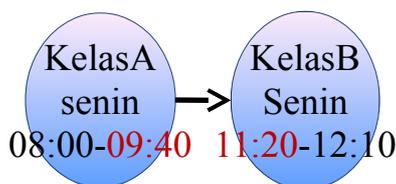
3. Menentukan bobot setiap *edge* atau sisi pada *graph*

Bobot graf terdiri dari dua jenis bobot, yaitu bobot sebenarnya($g(n)$) dan bobot heuristik($h(n)$). Bobot sebenarnya merupakan bobot yang diberikan pada sisi yang menghubungkan setiap simpul. Bobot ini ditentukan oleh beberapa kriteria, yaitu:

a. Sisa waktu terbang

Pembobotan yang dilakukan melalui perhitungan banyaknya rongga waktu terbang ketika kelas praktikum tersebut dijadwalkan dengan persamaan:

$$bobot = sisa\ SKS * 2 \tag{3}$$

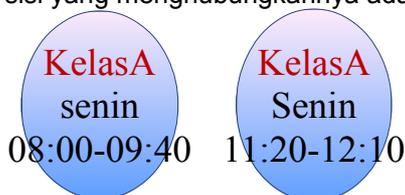


Gambar 5. Contoh perhitungan bobot berdasarkan sisa waktu terbang

Dari gambar diatas, terlihat bahwa sisa SKS antara kelas A dan kelas B adalah 100 menit(2 SKS). Sehingga bobot sisi tersebut berdasarkan kategori ini adalah 4.

b. Kelas yang berulang

Ketika dua simpul yang terhubung memiliki kelas praktikum yang sama meskipun memiliki waktu yang berbeda, maka bobot sisi yang menghubungkannya adalah 5. Selain itu diberi bobot 0.



Gambar 6. Contoh perhitungan bobot berdasarkan kelas berulang

Dari gambar diatas, terlihat bahwa kedua simpul tersebut merupakan kelas yang berulang, maka bobot sisi yang menghubungkannya diberi nilai 5.

c. Status dosen atau pengajar yang sudah terjadwal di jam yang sama.

Ketika pengajar pada simpul tujuan sudah pernah di jadwalkan sebelumnya pada jam yang sama di ruang yang lain (untuk pengulangan kedua dan seterusnya), maka sisi kedua simpul tersebut akan diberi bobot 100. Selain itu, maka diberi bobot 0.

Bobot heuristik merupakan bobot setiap simpul yang menggambarkan jarak perkiraan antara simpul n dengan simpul *goal* atau *start*. Sehingga pada metode MBDA* yang proses penelusurannya dilakukan secara dua arah, maka bobot heuristik terdiri dari $h_s(n)$ dan $h_g(n)$. $h_s(n)$ adalah biaya perkiraan dari simpul n ke simpul *goal*, dan $h_g(n)$ adalah biaya perkiraan dari simpul n ke simpul *start* [9].

Bobot heuristik dihitung dengan menggunakan fungsi *euclidean distance*. *Euclidean distance* merupakan jarak garis lurus antara dua buah vektor yang didefinisikan dengan suatu koordinat lokasi sumbu x dan y . Bobot heuristik pada proses penjadwalan dihitung dengan persamaan:

$$h(n) = \sqrt{(x.n - x.tujuan)^2 + (y.n - y.tujuan)^2} \tag{4}$$

Dimana :

- $x.tujuan$ = jumlah sks kelas n .
- $x.n$ = jumlah slot waktu kelas n
- $y.n$ = jumlah slot waktu / jumlah sks kelas n .

y.tujuan = 0.

4. Penelusuran jalur terpendek dengan algoritma MBDA*

Jalur terpendek pada *graph* akan menjadi solusi dalam proses penjadwalan. Jalur terpendek ditentukan berdasarkan total bobot jalur yang ditelusuri dan dihitung dengan persamaan MBDA yang telah dijelaskan sebelumnya. Setiap jalur terpendek yang ditelusuri merupakan solusi jadwal untuk setiap ruangan yang tersedia pada laboratorium.

Pada penerapannya, metode ini memiliki 2 buah larik utama, yaitu OPEN dan CLOSE. OPEN merupakan larik penyimpanan simpul yang akan menguraikan anak-anaknya untuk ditelusuri. Sementara itu CLOSED merupakan larik penyimpanan simpul terbaik sebagai calon solusi jalur terpendek. Sedangkan BestNode merupakan variabel penyimpanan simpul dengan bobot terbaik yang telah terpilih sebagai solusi jalur terpendek.

5. Membersihkan data simpul dari simpul yang sudah terjadwal

Ketika solusi untuk ruangan pertama didapatkan, maka pada saat itu *graph* harus bebas dari simpul yang kelas praktikumnya sudah terjadwal. Hal ini bertujuan untuk meminimasi *memory* penyimpanan data *graph* sekaligus untuk mempersingkat pencarian solusi untuk ruangan selanjutnya.

6. Mengulangi langkah 1-5 sebagai solusi akhir jadwal

Pengulangan kelima langkah diatas dilakukan sebanyak jumlah ruangan pada laboratorium. Karena setiap pengulangan akan menghasilkan solusi jadwal untuk satu ruangan.

3. Hasil dan Analisa

3.1. Pengujian Algoritma

Pengujian algoritma merupakan pengujian yang bertujuan untuk mengetahui tingkatan kinerja metode MBDA dalam melakukan proses penjadwalan. Pengujian tersebut dilakukan terhadap hasil penjadwalan dari 10 sampel kasus. Hasil penjadwalan dari setiap kasus akan dianalisa berdasarkan tiga kategori, yaitu Tidak bentrok, sesuai waktu kosong, dan optimal. Kategori "Tidak bentrok" merupakan kategori yang menentukan apakah jadwal yang dihasilkan tidak terdapat bentrokan waktu pengajar sehingga solusi dapat diterima. Sementara itu, Kategori "Sesuai waktu kosong" merupakan kategori yang menentukan apakah jadwal yang dihasilkan sesuai dengan waktu kosong pengajar dan mahasiswa yang diinputkan. Kategori "Optimal" merupakan kategori yang menentukan apakah jadwal yang dihasilkan merupakan solusi yang terbaik. Suatu jadwal dapat dikatakan "tidak optimal", jika terdapat suatu kelas praktikum yang masih bisa diletakkan pada suatu slot waktu, namun jadwal tersebut mengabaikannya.

Salah satu contoh kasus dilampirkan pada tabel berikut:

Tabel 1. Contoh kasus

No	Kelas Praktikum	SKS	Pengajar	Waktu kosong mahasiswa
1	Keamanan Komputer A	3	Nazruddin Safaat	Senin(08:00-10:30), Jum'at(13:00-15:30)
2	Keamanan Komputer B	3	Nazruddin Safaat	Senin(08:00-12:10), Selasa(08:00-12:10)
3	Sistem Digital C	3	Udin	Rabu(08:00-13:50)
4	Sistem Digital A	3	Sukir	Rabu(10:30-13:50)
5	Sistem Digital D	3	Sukir, udin	Rabu(08:00-10:30)
6	Sistem Digital B	3	Bejo	Rabu(08:00-13:50)
7	Struktur Data A	2	Lestari Handayani	Senin(08:00-12:10), Selasa(08:00-15:30), Rabu(10:30-12:10)
8	Daspro IA	3	Elin Haerani	Selasa(10:30-12:10), Senin(08:00-10:30), Rabu(08:00-12:10)
9	Praktikum Daspro IB	2	Sukir, Udin	Selasa(10:30-12:10), Kamis(08:00-10:30)

Dari tabel diatas terlihat bahwa setiap kelas praktikum memiliki pengajar yang berbeda-beda. Mahasiswa setiap kelas praktikum juga memiliki waktu kosong yang berbeda-beda. Pada waktu kosong tersebut sistem akan mencocokkan terhadap waktu kosong pengajar masing-masing kelas

yang kemudian sistem akan menyusun jadwal berdasarkan waktu tersebut. Data waktu kosong setiap pengajar dilampirkan pada tabel 2.

Tabel 2. Waktu kosong pengajar

No	Pengajar	Waktu kosong
1	Udin	Senin(09:40-12:10), Selasa(10:30-12:10), Rabu(13:00-15:30), Rabu(08:00-13:50), Jum'at(09:40-12:10)
2	Sukir	Selasa(08:00-12:10), Rabu(08:00-10:30), Rabu(10:30-14:40)
3	Nazruddin Safaat	Selasa(08:00-10:30), Senin(08:00-15:30)
4	Lestari Handayani	Selasa(08:00-12:10), Senin(08:00-10:30)
5	Elin Haerani	Senin(08:00-12:10), Selasa(10:30-14:40)
6	Bejo	Senin(08:00-11:20), Selasa(11:20-12:10), Rabu(08:00-10:30)

Dari data masukan seperti contoh kasus diatas, sistem akan melakukan penyusunan jadwal. Selanjutnya hasil penjadwalan dilampirkan pada tabel 3.

Tabel 3. Hasil akhir penjadwalan

Hari	Waktu	Ruang 1		Ruang 2	
		Kelas Praktikum	Pengajar	Kelas Praktikum	Pengajar
Senin	08:00-08:50	Daspro 1A	Elin Haerani	Keamanan Komputer A	Nazruddin Safaat
	08:50-09:40	Daspro 1A		Keamanan Komputer A	
	09:40-10:30	Daspro 1A		Keamanan Komputer A	
Selasa	08:00-08:50	Keamanan Komputer B	Nazruddin Safaat	Struktur Data A	Lestari Handayani
	08:50-09:40	Keamanan Komputer B		Struktur Data A	
	09:40-10:30	Keamanan Komputer B			
	10:30-11:20	Praktikum Daspro IB	Sukir, Udin		
	11:20-12:10	Praktikum Daspro IB			
Rabu	08:00-08:50	Sistem Digital B	Bejo	Sistem Digital D	Sukir, udin
	08:50-09:40	Sistem Digital B		Sistem Digital D	
	09:40-10:30	Sistem Digital B		Sistem Digital D	
	10:30-11:20	Sistem Digital A	Sukir	Sistem Digital C	udin
	11:20-12:10	Sistem Digital A		Sistem Digital C	
	12:10-13:00	Sistem Digital A		Sistem Digital C	

Hasil pengujian algoritma terlampir pada tabel berikut.

Tabel 4. Hasil pengujian algoritma

Kasus	Bentrokkan	Sesuai waktu kosong	Optimal
Kasus 1	Tanpa bentrokkan	Sesuai	Optimal
Kasus 2	Tanpa bentrokkan	Sesuai	Tidak optimal

Kasus 3	Tanpa bentrokan	Sesuai	Optimal
Kasus 4	Tanpa bentrokan	Sesuai	Tidak optimal
Kasus 5	Tanpa bentrokan	Sesuai	Tidak optimal
Kasus 6	Tanpa bentrokan	Sesuai	Tidak optimal
Kasus 7	Tanpa bentrokan	Sesuai	Tidak optimal
Kasus 8	Tanpa bentrokan	Sesuai	Optimal
Kasus 9	Tanpa bentrokan	Sesuai	Optimal
Kasus 10	Tanpa bentrokan	Sesuai	Optimal

Dari tabel diatas terlihat bahwa seluruh kasus pada pengujian dapat menghasilkan jadwal yang baik tanpa adanya bentrokan dan sesuai dengan waktu kosong yang diajukan oleh mahasiswa dan pengajar. Namun, diantara 10 kasus pengujian terdapat 5 kasus, yaitu kasus 2,4,5,6, dan 7 yang menghasilkan jadwal yang tidak optimal. Jadwal yang tidak optimal dikarenakan terdapatnya kelas praktikum yang tidak dijadwalkan, sementara slot waktu masih tersedia.

3.2. Pengujian *User Acceptance Test*

Pada pengujian ini, kuisisioner diajukan kepada 4 jenis *user* yang akan menggunakan sistem. *User* tersebut adalah 1 orang kepala laboratorium, 4 orang dosen, 4 orang asisten dosen, dan 1 orang laboran. Pertanyaan untuk setiap jenis *user* tidak seluruhnya sama. Terdapat pertanyaan yang berbeda untuk setiap jenis *user*. Berdasarkan pengujian *User Acceptance Test*, didapatkan beberapa kesimpulan berdasarkan 3 kategori yang diuji:

- Penilaian terhadap *interface* sistem menyatakan bahwa 50% pengguna memberikan penilaian “baik” dan 50% memberikan penilaian “sangat baik” terhadap *interface* sistem.
- Penilaian terhadap tepat sasaran sistem menyatakan bahwa 80% pengguna memberikan penilaian “baik” dan 20% memberikan penilaian “sangat baik”

4. Kesimpulan

Berdasarkan pengembangan sistem penjadwalan ini, dapat diambil beberapa kesimpulan diantaranya adalah:

- Metode MBDA* dapat di terapkan untuk menyelesaikan kasus penjadwalan, khususnya penjadwalan penggunaan laboratorium
- Perhitungan bobot *graph* setiap kelas dihitung berdasarkan 3 kategori, yaitu sisa waktu yang terbuang, kelas yang berulang, dan sudah terjadwal diruang lain. Dari ketiga kategori tersebut, seluruh data yang diujikan menghasilkan jadwal yang bebas dari bentrokan waktu pengajar dan ruang laboratorium(100%). Seluruh data pengujian juga menghasilkan jadwal yang sesuai dengan waktu kosong yang diajukan pengguna laboratorium(100%).
- Berdasarkan percobaan 10 kasus secara acak, proses penyusunan jadwal menghasilkan 5 jadwal yang merupakan solusi optimal yang berarti sistem dapat menjadwalkan seluruh kelas praktikum yang harusnya mendapatkan slot waktu.

Referensi

- [1] Hasanah, S. Aplikasi pewarnaan Graph terhadap Penjadwalan Kuliah di Jurusan Matematika UIN Malang. Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi UIN Malang. 2007.
- [2] Yustina E. Design dan Implementasi Jadwal Kuliah dengan Menggunakan Algoritma Semut Berbasis Web. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Malang. Malang. 2008

- [3] Selpira P. Sistem Penjadwalan Dokter Jaga Menggunakan Algoritma Greedy dengan Permutasi. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN SUSKA. Pekanbaru. 2009
- [4] Oktaviani, W. Membangun Aplikasi Penjadwalan Asisten Praktikum di STMIK AMIKOM YOGYAKARTA. Yogyakarta. 2011.
- [5] Suyanto. *Artificial Intelligence (Edisi Revisi)*. Informatika. Bandung. 2007.
- [6] Dechter R, Judea P. *Generalized best-first search strategies and the optimality of A**. *Journal of the ACM* 32 (3): pp. 505 – 536. 1985
- [7] Adipranata R, Handojo A, dan Setiawan H. Aplikasi Pencari Rute Optimum Pada Peta Guna Meningkatkan Efisiensi Waktu Tempuh Pengguna Jalan Dengan Metode A* dan Best First Search. Laporan Penelitian. Universitas Kristen Petra. Surabaya. 2007. James S, Whales D. *The Framework of Electronic Government*. U.S. Dept. of Information Technology. Report number: 63. 2005.