

Course Scheduling Using Dynamic-Order Backjumping (Case Study: STIKA Madiun)

Muh Zaenal Abidin
Informatika, Fakultas MIPA
Universitas Sebelas Maret
Jalan Ir. Sutami 36A Surakarta
zainal.abidin37@student.uns.ac.id

Wiranto
Informatika, Fakultas MIPA
Universitas Sebelas Maret
Jalan Ir. Sutami 36A Surakarta
wiranto@staff.uns.ac.id

Haryono Setiadi
Informatika, Fakultas MIPA
Universitas Sebelas Maret
Jalan Ir. Sutami 36A Surakarta
hsd@staff.uns.ac.id

Abstract

Scheduling is one of constraint satisfaction problem (CSP). CSP could be solved by Backtracking (BT) algorithm. BT algorithm usually searches for already known wrong node or thrashing. To reduce thrashing on BT, we can use Backjumping (BJ) algorithm. In BT and ordinary BJ, they form a search tree that has fixed sequential variables. This order of variables has an effect on how much node created. Bad order could increase thrashing. Dynamic-order Backjumping (DOBJ) algorithm uses dynamic sequence variables to reduce thrashing. In the implementation, this research used data from the Science College of Train (STIKA) Madiun during 2015/2016 academic year on second semester. Data used in this research, including majors, lecturers, courses, and hour choice. The limits in this research only used hard constraint. From one of a result test, DOBJ has shorten runtime by making 29 times BJ became 5 times order change and 28 times BT. Because of the changes in varying order, it could change BJ step to BT step, so the execution time would be faster.

Keyword: CSP, scheduling, backtracking, backjumping, dynamic-order backjumping

1. Pendahuluan

Penjadwalan mata kuliah menjadi sesuatu yang penting. Apabila batasan-batasan penjadwalan tidak diperhitungkan dengan baik maka sulit ditemukannya penjadwalan yang sesuai dengan kebutuhan [1]. Penggunaan ruang, waktu dan kebutuhan mahasiswa perlu diperhatikan agar terhindar dari permasalahan tabrakan antara mata kuliah, tenaga pendidik dan waktu yang diakibatkan faktor penentu (*constraints*) [2]

Sekolah Tinggi Ilmu Kereta Api (STIKA) Madiun memiliki pola pengaturan jadwal mata kuliah yang masih manual. Hal tersebut tentu menyita banyak waktu. Oleh karena itu, diperlukan sistem penjadwalan mata kuliah untuk menyusun penjadwalan berdasarkan batasan atau kendala tersebut.

Dalam penjadwalan, para peneliti menggunakan beberapa macam metode, seperti algoritma genetika [1] [2] [3] dan algoritma *backtrack* (BT) [4]. Penggunaan algoritma genetika oleh Adi Chandra Syarif dan Farif Hartono Gunawan, yang mengesimpulkan bahwa kompleksitas permasalahan bertambah dengan bertambahnya batasan [2]. Penggunaan algoritma genetika dalam penjadwalan, memakan waktu yang lama untuk membuat jadwal dan hasil yang dihasilkan tidak memuaskan [3]. Pada penelitian oleh Rosni Lumbantoruan, et al. menggunakan algoritma BT untuk

membuat sistem penjadwalan memiliki waktu yang relatif singkat, dan tidak ada konflik [4]. Dan pada penelitian oleh Siswono Teno, Palgunadi Sarngadi dan Afrizal Doewes mengkombinasikan algoritma genetika dan algoritma palgunadi untuk mengurangi waktu eksekusi pada algoritma genetika [3].

Pada penelitian ini berfokus pada BT. BT biasanya hanya melakukan satu langkah kedepan maupun kebelakang, dapat melakukan pencarian simpul yang sudah pasti salah (*thrashing*) [5]. Berdasarkan hal tersebut, untuk mengurangi *thrashing*, penelitian ini menggunakan turunan BT yaitu algoritma *Backjumping* (BJ). BJ merupakan turunan dari BT untuk mengurangi *thrashing*. Karena BT dapat bekerja lebih baik apabila urutan variabelnya lebih baik [6]. Untuk menggunakan waktu pemrosesan secara efisien dalam penelitian ini menggunakan menggunakan algoritma *backjumping* dengan pengurutan dinamis.

2. Tinjauan Pustaka

2.1. Constraint Satisfaction Problem (CSP)

Constraint satisfaction problem (CSP) merupakan sebuah masalah penempatan set variabel dalam sebuah domain. Sebuah CSP atau *constraint network* $P = (X, D, C)$ terdiri dari satu set variable $X = \{x_1, \dots, x_n\}$, satu set domain $D = \{d_1, \dots, d_n\}$, dan satu set masalah $C = \{c_1, \dots, c_n\}$ [5]. Terdapat tiga komponen utama yang perlu diperhatikan dalam pendekatan CSP, yaitu [7]:

- Constraint*: merupakan suatu aturan yang ditentukan untuk mengatur nilai yang boleh diisikan ke dalam variabel. Terdapat beberapa jenis *constraint*, diantaranya *unary* (menyatakan persyaratan sebuah variabel), *binary* (menyatakan persyaratan sepasang variabel), *n-ary* (menyatakan persyaratan tiga atau lebih variabel), dan *preference* (syarat yang sebaiknya dipenuhi, tetapi tidak harus)
- Domain*: merupakan kumpulan nilai legal yang dapat diisi ke dalam variabel. Dengan kata lain, sebuah domain akan membatasi nilai suatu variabel.
- Variabel*: merupakan suatu penampung yang dapat diisi dengan berbagai nilai. Biasanya persoalan dimulai di sini, yaitu ketika variabel harus diisi oleh domain yang telah memenuhi *constraint*.

2.2. Jenis Constraint

Secara pemenuhan prioritasnya, *constraint* dapat dibedakan menjadi 2 bagian, yaitu [7] :

- Hard Constraint*: adalah persyaratan yang harus dipenuhi dan tidak boleh dilanggar dalam pembuatan penyelesaian masalah.

b. *Soft Constraint*: adalah persyaratan tambahan yang biasanya merupakan sebuah permintaan khusus, tanpa adanya konsekuensi fatal jika tidak terpenuhi.

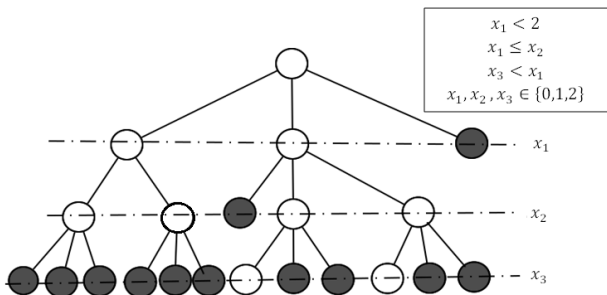
2.3. Algoritma Backtracking

Algoritma Backtracking pertama kali diperkenalkan oleh D.H. Lehmer, ahli matematika Amerika pada tahun 1950. Backtracking (BT) adalah algoritma umum untuk menemukan semua atau beberapa solusi untuk menyelesaikan masalah komputasi, terutama CSP, yang secara bertahap memilih kandidat solusi dan mengabaikan bagian solusi lain setelah diketahui kandidat tersebut tidak dapat menjadi bagian solusi [8]. Algoritma Backtracking merupakan algoritma pencarian primer dalam menyelesaikan CSP [5].

Langkah-langkah pencarian solusi dengan BT adalah sebagai berikut [9] :

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Simpul yang sudah dilahirkan dinamakan simpul hidup dan simpul hidup yang diperluas dinamakan simpul-E (*Expand-node*).
2. Jika lintasan yang diperoleh dari perluasan simpul-E tidak mengarah ke solusi, maka simpul itu akan menjadi simpul mati dimana simpul itu tidak akan diperluas lagi.
3. Jika posisi terakhir ada di simpul mati, maka pencarian dilakukan dengan membangkitkan simpul anak yang lainnya dan jika tidak ada simpul child maka dilakukan backtracking ke simpul parent
4. Pencarian dihentikan jika kita telah menemukan solusi atau tidak ada simpul hidup yang dapat di diperluas.

Dalam BT terbentuk sebuah pohon yang dicari secara rekursif. Pencarian akan berhenti jika solusi ditemukan, atau juga bisa dilanjutkan untuk mencari beberapa solusi.



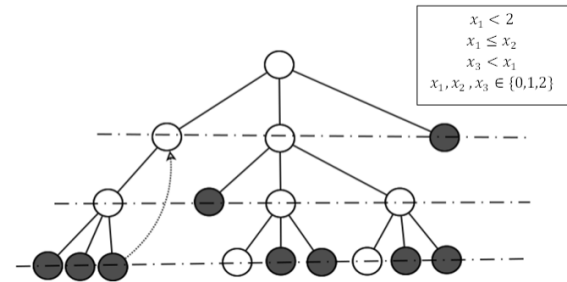
Gambar 1. Contoh pohon Backtracking

Dapat dilihat pada gambar 1 menunjukkan ketika $x_1 = 0$, tidak dapat menghasilkan nilai karena $x_3 < x_1$. Tetapi dalam BT akan tetap menelusuri pada cabangnya meskipun jelas tidak ada solusi, atau biasa disebut *thrashing*.

2.4. Algoritma Backjumping

Pada Backtracking, jika x_i tidak berkaitan dengan x_{i+1} maka cabang dari x_{i+1} dapat dipastikan tidak dapat digunakan, namun dalam backtracking semua cabangnya akan ditelusuri sehingga memakan waktu yang sia-sia (*thrashing*). Prosedur *Backjumping* oleh Gaschnig (1979) mencoba untuk mengurangi simpul yang dikunjungi pada pohon pencarian dan secara konsekuen mengurangi

konsistensi cek yang dilakukan oleh proses pencarian [10].

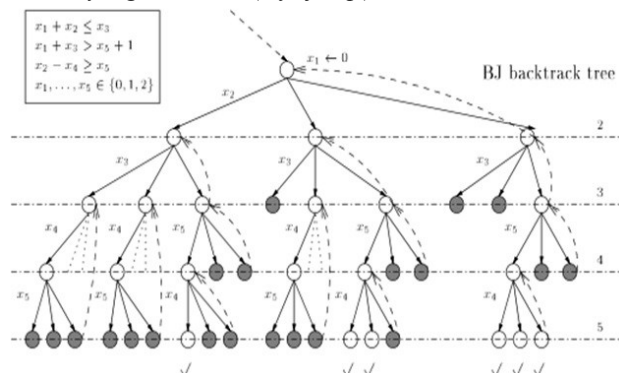


Gambar 2. Contoh pohon Backjumping

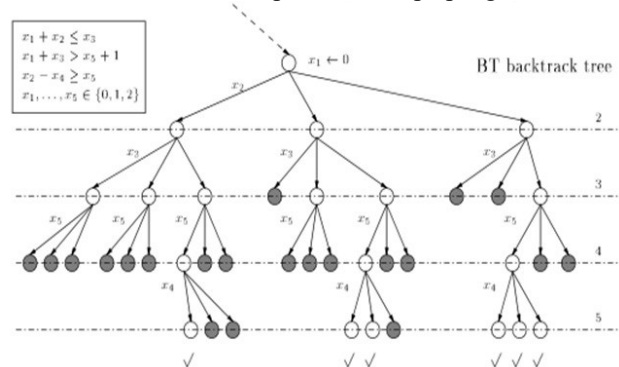
Pada gambar 2 ketika $\{x_1 = 0, x_3 = 0\}$, x_3 tidak memberikan solusi maka akan dilakukan pengecekan apakah tanpa solusi pada x_2 nilai x_3 dapat ditemukan. Karena tidak ada nilai untuk x_3 maka melompat kembali (*backjumping*) ke x_1 tanpa mengecek nilai selanjutnya untuk x_2 , lalu diberikan nilai selanjutnya untuk x_1 .

2.5. Urutan Variabel pada Pohon Kesimpulan

Menurut Chen dan van Beek (2001), pada pengurutan variabel pada *backtrack* akan memiliki simpul yang lebih sedikit dari pada *Conflict-Directed Backjumping* (CBJ) tanpa pengurutan. CBJ hampir sama seperti BJ tetapi dengan pencatatan konflik, sehingga bisa melompat pada konflik yang berkaitan (*safe jump*).



Gambar 3. Ilustrasi pohon BJ tanpa pengurutan



Gambar 4. Ilustrasi pengurutan variabel untuk BT dari pohon BJ

Pada gambar 3 dan gambar 4 menunjukkan BJ tidak bisa mengunjungi simpul lebih sedikit dibanding BT yang sudah diurutkan. Dan apabila BJ atau CBJ diterapkan pada pohon ke 2, mereka juga tidak lebih sedikit mengunjungi simpul dibanding BT.

3. Metodologi

3.1. Pemodelan dan Perancangan

3.1.1. Pemodelan Data

Pemodelan yang digunakan dalam sistem berdasarkan kebutuhan penjadwalan pada STIKA Madiun, sebagai berikut:

- a. Dosen merupakan set dari dosen, $D = \{d_1, d_2, \dots, d_n\}$.
- b. Kelas merupakan set dari kelas, $K = \{k_1, k_2, \dots, k_n\}$.
- c. Jumlah jam pelajaran tiap mata kuliah, merupakan set dari jam pelajaran, $P = \{p_1, p_2, \dots, p_n\}$.
- d. Jam pelajaran disini tidak menggunakan jam dunia (24 jam sehari), tetapi menggunakan jam pelajaran, sehingga waktu istirahat (sholat, makan, dll) tidak berpengaruh pada penjadwalan. Satuan jam pelajaran, $t = \{t_1, t_2, \dots, t_n\}$.
- e. Hari merupakan set hari, $h = \{h_1, h_2, \dots, h_n\}$.
- f. Timeslot merupakan gabungan hari dan jam.
- g. Mata kuliah merupakan set dari mata kuliah, $M = \{m_1, m_2, \dots, m_n\}$, digunakan variabel utama yang dicari solusi penempatannya, sehingga setiap mata kuliah memiliki beberapa dosen, dan jumlah jam pelajaran. $m_i = \{d_j, d_{j+1}, \dots, p_k\}$
- h. Jadwal merupakan set dari jadwal, $J = \{j_1, j_2, \dots, j_n\}$ memiliki atribut hari, jam, kelas dan mata kuliah. Sehingga satu jadwal merupakan, $j_z = \{h_y, t_x, k_w, m_v\}$

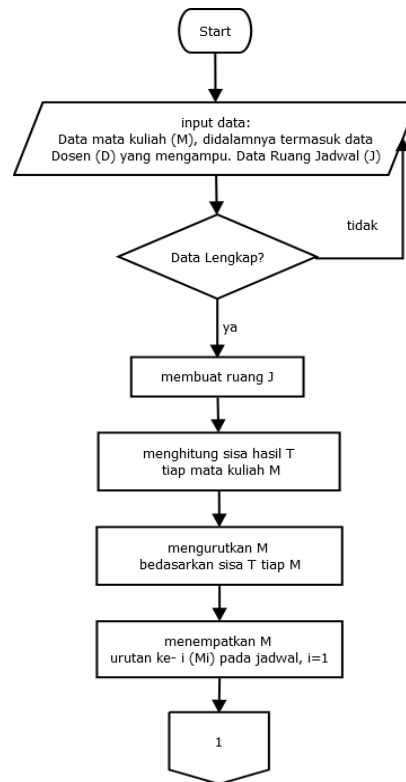
3.1.2. Batasan Penjadwalan

Dalam penelitian ini menggunakan menggunakan batasan yang tidak boleh dilanggar atau batasan kaku (*Hard Constraint*), yaitu:

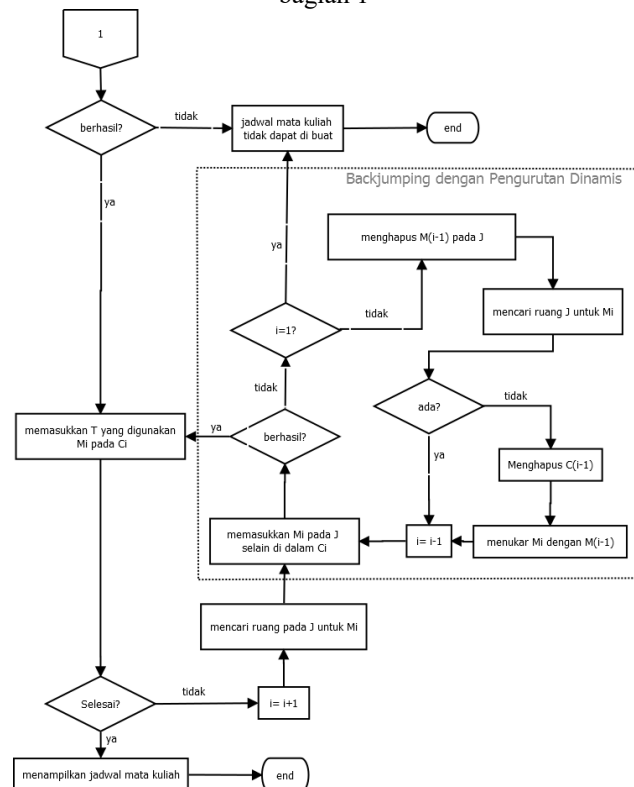
- a. Setiap dosen hanya mengajar 1 mata kuliah dalam 1 timeslot
- b. Dalam 1 kelas dan 1 timeslot hanya ada 1 mata kuliah
- c. Dosen memiliki waktu – waktu yang tidak bisa digunakan (waktu pilihan dosen)
- d. Pilihan jam berdasarkan jam pelajaran pada 1 minggu dan tidak berubah-ubah
- e. Pilihan jam mata kuliah minimal sebanyak jam pelajaran mata kuliah tersebut
- f. Pilihan jam dosen minimal sebanyak total jam pelajaran mata kuliah yang diampu
- g. Pada mata kuliah dengan hari dan kelas yang sama, jam pelajaran harus berurutan
- h. Total jam pelajaran tidak melebihi alokasi jam pada 1 minggu

3.1.3. Perancangan Sistem

Penjadwalan diselesaikan dengan menggunakan Algoritma *Backjumping*. Perancangan sistem menggunakan Bahasa pemrograman PHP dengan framework Codeigniter. Inputan sistem terdiri dari: nama mata kuliah, jumlah jam pelajaran, semester pada mata kuliah, nama dosen, data prodi, jumlah ruang, nama dosen, data dosen yang mengampu mata kuliah, pilihan hari pada dosen, pilihan hari pada mata kuliah. Kemudian hasil yang diberikan oleh sistem berupa jadwal mata kuliah. Adapun langkah-langkah algoritma digambarkan ke dalam *flowchart* seperti pada Gambar 4.



Gambar 4. Flowchart *backjumping* dengan urutan dinamis bagian 1



Gambar 5. Flowchart *backjumping* dengan urutan dinamis bagian 2

Sistem penjadwalan dapat berjalan setelah input dimasukkan. Untuk melakukan pencarian dibuat ruang pencarian, yaitu ruang jadwal. Ruang jadwal berdasarkan hari, jam, dan kelas. Data input yang dimasukkan dapat menentukan jumlah kemungkinan ruang jadwal yang dapat di isi oleh mata kuliah. Kemudian setelah diketahui jumlah kemungkinan dilakukan pengurutan. Pengurutan

kemungkinan berdasarkan dari yang paling sedikit, sehingga konflik dapat dideteksi lebih dini.

Langkah berikutnya yaitu mencoba kemungkinan pada mata kuliah. Mata kuliah ditempatkan pada salah satu kemungkinan dan diikuti oleh mata kuliah lain. Ketika tidak ditemukan kemungkinan pada mata kuliah maka akan melakukan cek mata kuliah yang konflik. Pegecekan dilakukan dengan cara, mengecek apakah mata kuliah sebelumnya berpengaruh pada ruang solusi. Dalam sistem pengecekan dilakukan dengan mengabaikan 1 hingga beberapa mata kuliah dalam ruang penjadwalan. Setelah diketahui mata kuliah yang konflik, maka mata kuliah tersebut ditempatkan di bawahnya, sehingga mata kuliah yang konflik berurutan. Penempatan ruang jadwal akan terus berlanjut sampai semua mata kuliah dimasukkan, atau terjadi *error* karena tidak ada jadwal yang mungkin dilakukan.

4. Hasil dan Pembahasan

Data yang digunakan merupakan data mata kuliah dan dosen di STIKA Madiun tahun ajaran 2015/2016 semester ganjil. Data yang menggunakan data pada 4 (empat) program studi: Program Studi Diploma III Teknik Bangunan dan Jalur Perkeretaapian (TBJP), Diploma III Teknik Elektro Perkeretaapian (TEP), Diploma III Teknik Mekanika Perkeretaapian (TMP), dan Diploma III Manajemen Transportasi Perkeretaapian (MTP). Tabel 1 merupakan ringkasan data Mata Kuliah yang digunakan.

Tabel 1. Ringkuman data matakuliah semester ganjil pada tahun ajaran 2015/2016

Jurusan	Total Kelas	Total Mata Kuliah	Total Jam Pelajaran
TBJP	3	17	74
TEP	3	19	61
TMP	3	20	72
MTP	3	18	70
Total	12	74	277

Pada setiap jurusan, setiap tingkat memiliki 1 kelas. Setiap kelas memiliki beberapa mata kuliah. Setiap mata kuliah memiliki jumlah jam pelajaran (JP) untuk dosen dan untuk asisten dosen (asdos). Total dosen (termasuk asdos) yang mengajar 4 program studi tersebut berjumlah 61 orang. Setiap mata kuliah dapat ditentukan jam pelajaran yang dipilih. Dan setiap dosen juga dapat mengambil jam – jam tertentu sesuai kebutuhan. Contoh data mata kuliah yang digunakan dapat dilihat pada Tabel 2. Untuk dosen, dalam penelitian ini menggunakan id dan nama dosen.

Tabel 2. Contoh data tabel mata kuliah

No	Semester	Mata Kuliah	Jumlah Dosen	JP Dosen	Jumlah Asdos	JP Asdos	Jurusan
1	1	Bahasa Indonesia	1	2	1	2	TEP

No	Semester	Mata Kuliah	Jumlah Dosen	JP Dosen	Jumlah Asdos	JP Asdos	Jurusan
2	1	Bahasa Inggris	1	3	2	2	TEP
3	1	Dasar Menggambar Teknik	1	3	2	2	TEP
4	1	Fisika	1	4	2	2	TEP
5	1	Kewarganegaraan	1	2	1	2	TEP

Simpul menggunakan dasar mata kuliah. Mata kuliah dibagi menjadi 2, yaitu teori dan praktikum. Pembagian teori dan praktikum pada matakuliah ditentukan oleh jumlah JP dosen dan JP asdos. Misal, jika mata kuliah X memiliki 5 JP dosen dan 4 JP asdos, maka mata kuliah X akan dibagi menjadi mata kuliah X teori dengan 1 JP dan praktikum dengan 2 JP. Pada mata kuliah teori hanya terdapat dosen, dan pada mata kuliah praktikum terdapat dosen dan asdos. Setiap mata kuliah dan dosen memiliki jam pilihan. Kemudian pilihan jam dosen (atau sebagai asdos) dimasukkan pada setiap mata kuliah yang diampu.

$$x_i = m_i \cap d_m$$

x : Pilihan jam simpul

m : Pilihan jam mata kuliah

d : Pilihan jam dosen yang mengampu mata kuliah ke- i .

(Apabila terdapat dosen lain yang mengampu, maka d merupakan *intersection* dari semua dosen yang mengampu mata kuliah ke- i)

Dalam pengujian dilakukan beberapa tes untuk *backtrack* (BT), *backjumping*(BJ), dan *backjumping* dengan pengurutan dinamis /*dynamic order backjumping* (DOBJ). Pengujian dilakukan dengan batas maksimal (*threshold*) 5 jam pemrosesan. Pengujian dilakukan pada komputer dengan spesifikasi sebagai berikut.

Hardware :

- Processor : AMD Phenom II X4 945 (4 CPUs), 3 GHz
- RAM : 4096 MB

Software :

- Windows 10 Enterprise 64-bit
- XAMPP control panel v3.2.2
- APACHE 2.4.27 / PHP 7.1
- MySQL 5.0

Pengujian dilakukan dengan 5 kondisi, dengan menentukan pilihan jam pada dosen dan atau pada mata kuliah. Pada penelitian ini pengujian berfokus pada pilihan jam dosen karena, mata kuliah sebagai dasar simpul sehingga dengan dipilihnya pilihan jam dosen yang dapat berdampak pada beberapa mata kuliah atau simpul.

Gambar 6. Contoh hasil penjadwalan

Tabel 3. Hasil Pengujian *Backtrack* (BT), *Backjumping* (BJ), *Backjumping* dengan pengurutan dinamis / *Dynamic Order Backjumping* (DOBJ)

Nama	Pilihan Jam		Hasil		
	Dosen	Matakuliah	BT	BJ	DOBJ
Test 1	- Tidak ada	- Tidak ada	- 28,75 detik - 133 langkah - 0 BT	- 29,19 detik - 0 BJ - 133 langkah - 0 bt	- 29,13 - 0 <i>swapping</i> - 133 langkah - 0 BT
Test 2	- Aditya Wahyu Erlangga (13), Selasa tidak bisa. - Surjanto(15), Kamis tidak bisa - Akbar Zulkarnain (16), Jumat tidak bisa	- Tidak ada	- 15808,36 detik (4 jam 23 menit 28,36 detik) - 159785 langkah - 53217 BT	- 28,77 detik - 1 BJ (2 level) - 0 BT - 139 langkah	- 29,40 detik - 2 <i>swapping</i> - 0 BT - 139 langkah
Test 3	- Arisma Havino Wantana(7) dan Septiana Widi Astuti (12), Senin tidak bisa. - Dadang Sanjaya(8) dan Aditya Wahyu Erlangga(13), Selasa tidak bisa. - Sunaryo(9), Rabu tidak bisa - Surjanto(15), Kamis tidak bisa - Akbar Zulkarnain (16), Jumat tidak bisa.	- Tidak ada	- Lebih dari <i>threshold</i>	- 31,33 detik - 1 BJ (4 level) 1 BJ (2 level) - 0 BT - 149 langkah	- 30,83 detik - 4 <i>swapping</i> - 0 BT - 144 langkah
Test 4	- Arisma Havino Wantana(7) dan Septiana Widi Astuti (12), Senin tidak bisa. - Dadang Sanjaya(8) dan Aditya Wahyu Erlangga(13), Selasa tidak bisa. - Arisma Havino Wantana(7), Sunaryo(9), Rabu tidak bisa. - Dadang Sanjaya(8),Surjanto(15), dan Erifendi Churniawan(17), Kamis tidak bisa. - Arisma Havino Wantana(7) Akbar Zulkarnain (16), dan Setia Hadi Pramudi(18), Jumat tidak bisa.	- pengantar sarana perekertaapia n (TEP) rabu jam 7, 8 tidak bisa	- Lebih dari <i>threshold</i>	- 76,15 detik (1 menit 16,15 detik) - 29 BJ (4 level) - 0 BT - 149 langkah	- 39,58 detik - 5 <i>swapping</i> - 28 BT - 149 langkah

Nama	Pilihan Jam		Hasil		
	Dosen	Matakuliah	BT	BJ	DOBJ
Test 5	<ul style="list-style-type: none"> - Arisma Havino Wantana(7), Septiana Widi Astuti (12), Ima Widinayah(21), Senin tidak bisa - Dadang Sanjaya(8) dan Aditya Wahyu Erlangga(13), Catur Wicaksono(27), Selasa tidak bisa - Arisma Havino Wantana(7) dan Sunaryo(9), Rabu tidak bisa. - Dadang Sanjaya(8),Surjanto(15), dan Erifendi Churniawan(17),Sachiko Mawadah(22), Kamis tidak bisa - Arisma Havino Wantana(7) Akbar Zulkarnain (16), dan Setia Hadi Pramudi(18), Jumat tidak bisa 	<ul style="list-style-type: none"> - pengantar sarana perekertaaipian (TEP) rabu jam 7, 8 tidak bisa - Dasar Operasi Kereta Api(TBJP) dan Bahasa Inggris II (TEP) senin tidak bisa 	<ul style="list-style-type: none"> - 30,26 detik - 0 BT - 133 langkah 	<ul style="list-style-type: none"> - 30,30 detik - 0 BJ - 0 BT - 133 langkah 	<ul style="list-style-type: none"> - 29,24 detik - 0 <i>swapping</i> - 0 BT - 133 langkah

Hasil pengujian dapat dilihat pada Tabel 3. *Backjumping* (BJ) dan *Backjumping* dengan pengurutan dinamis (DOBJ) masih menggunakan *Backtrack* (BT). Pada BJ memiliki level yang berarti jumlah simpul induk yang dilewati karena melakukan *trashing*. Sedangkan pada DOBJ langkah BJ digantikan dengan *swaping* yang berarti membalikkan urutan simpul dengan simpul induk satu level. Pada pengujian didapatkan kesimpulan sebagai berikut:

- a. Pada Test 1 menunjukkan running time yang relatif sama antara ketiga algoritma tersebut.
- b. Pada Test 2 dan 3 menunjukkan bahwa BJ dan DOBJ memiliki waktu relatif sama dan memiliki selisih yang besar dengan *backtrack*, ini menunjukkan dengan mengurangi *trashing* dapat berdampak signifikan pada pencarian solusi.
- c. Pada Test 4 menunjukkan bahwa dengan mengubah urutan simpul dapat mengurangi waktu eksekusi karena simpul konflik berdekatan, dan tidak perlu melakukan *Backjumping*.
- d. Pada Test 5 menunjukkan dengan banyaknya *constraint* bukan berarti semakin mendekati ke *worst-case*. Jika urutan sesuai dapat mendekati ke *best case*.

5. Kesimpulan Dan Saran

Dari hasil pengujian pada penelitin ini, algoritma *backjumping* dengan pengurutan dinamis dapat membantu menyelesaikan penjadwalan Sekolah Tinggi Ilmu Kereta Api (STIKA) Madiun. Pada pengujian dimana *thrashing* dihindari oleh *Backjumping* (BJ) dan *Backjumping* dengan Pengurutan Dinamis / Dynamic Order *Backjumping* (DOBJ), ketika keduanya menggunakan 137 langkah dengan waktu BJ 30 detik dan DOBJ 29 detik. Pada pengujian yang sama *Backtracking* (BT) menggunakan 159783 langkah dengan waktu 15808 detik. Pengujian ini menunjukkan pengurangan *thrashing* dapat berdampak pada langkah eksekusi dan waktu eksekusi.

Pengujian pada kondisi dimana pada algoritma BJ melakukan *backjumping* sebanyak 29 kali pada simpul yang sama sehingga menggunakan running time 76 detik. Pada kondisi yang sama, DOBJ melakukan 5 kali *swapping* dan 28 *backtracking* dan menggunakan running time 40 detik. Dengan perubahan urutan

variabel yang dilakukan oleh DOBJ dapat meringkas langkah *backjumping* menjadi *backtracking* sehingga mempercepat waktu pemrosesan.

Berikut saran untuk penelitian selanjutnya:

1. Baik BJ maupun DOBJ masih memiliki *worst-case* yang sama dengan BT yaitu (n^m) sehingga ruang solusi sebaiknya diperkecil, dengan melakukan kombinasi algoritma lain.
2. DOBJ perlu ditingkatkan agar dapat melakukan *swaping* ketika *trashing* dilakukan oleh kombinasi 2 simpul atau lebih.

DAFTAR PUSTAKA

- [1] Chen, X., & van Beek, P. (2001). Conflict-Directed *Backjumping* Revisited. *Journal of Artificial Intellegence Research* 14, 53-81.
- [2] Dechter, R., & Frost, D. (2002). Constraint Networks. *Backjump-based backtracking for constraint satisfaction problems*, 147-188.
- [3] Knuth, D. E. (1968). *The Art of Computer Programming Volume II*. Addison-Wesley.
- [4] Lumbantoruan, R., Simatupang, Y. N., Siahaan, M. N., Pardede, M. H., & Pakpahan, J. (2012). Penjadwalan Kuliah dengan Algoritma *Backtracking*. *Konferensi Nasional ICT-M Telkom (KNIP)*, (hal. 255-264). Bandung.
- [5] Prosser, P. (1993). Hybrid algorithms for the constraint satisfaction problem. *Computational intelligence*, 9(3), 268-299.
- [6] Puspaningrum, W. A., Djunaidy, A., & Vinarti, R. A. (2013). Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. *JURNAL TEKNIK POMITS Vol. 2, No. 1, (2013) ISSN: 2337-3539 (2301-9271 Print)*, 1-5.
- [7] Russell, S., & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach (3rd ed)*.
- [8] Syarif, A. C., & Gunawan, F. H. (2013). Penerapan Algoritma Evolusi Dengan Metode Generation Replacement Pada Aplikasi Penjadwalan Mata Kuliah (Studi Kasus Universitas Atma Jaya Makassar). *JURNAL TEMATIKA VOL. 1, NO. 2*, 1-14.
- [9] Teneng, Purwadi, J., & Kurniawan, E. (2010, April 1). Penerapan Algoritma *Backtracking* pada

Permainan Math Maze. *Jurnal Informatika*, 6, 56-67.

- [10] Teno, S., Sarngadi, P., & Doewes, A. (2016). Kombinasi Algoritma Genetika dengan Algoritma Palgunadi untuk Penjadwalan Mata Kuliah di Universitas Sebelas Maret. *ITSmart: Jurnal Teknologi dan Informasi*, 2, 7-12.