

Implementasi Algoritma Gale – Shapley pada Situs Jejaring Sosial Pencarian Kerja UMN Vacancy

Ivan Bong

Program Studi Teknik Informatika
Fakultas Teknologi Informasi dan Komunikasi
Universitas Multimedia Nusantara
Gading Serpong, Tangerang, Indonesia
sivenvipes@gmail.com

Dodick Z. Sudirman S.Kom., B.App.Sc., M.T.I.

Program Studi Teknik Informatika
Fakultas Teknologi Informasi dan Komunikasi
Universitas Multimedia Nusantara
Gading Serpong, Tangerang, Indonesia
dodick@umn.ac.id

Abstrak—Departemen Pengembangan Karir Universitas Multimedia Nusantara (UMN) menyediakan wadah bagi mahasiswa untuk mencari dan memilih lowongan kerja yang sesuai dengan minat dan kemampuan mereka. Dalam proses pemasangan berdasarkan tanya jawab akan sulit menjamin apakah kombinasi mahasiswa dan lowongan kerja adalah yang paling baik karena kita tidak memiliki *preference list* dari kedua belah pihak. Oleh karena itu, penelitian ini berfokus pada bagaimana mengimplementasikan Algoritma Gale – Shapley pada situs jejaring sosial pencarian kerja di UMN yang bertujuan memasang pencari kerja dengan lowongan kerja yang diterbitkan sehingga pencari kerja mendapatkan pekerjaan yang sesuai dan diinginkan, dan perusahaan mendapatkan karyawan yang sesuai dengan klasifikasi lowongan kerja yang diterbitkan. Dari hasil uji coba *pairing* antara pencari kerja dan lowongan kerja, dapat disimpulkan bahwa Algoritma Gale – Shapley memenuhi prinsip *definiteness*, *input*, *output*, dan *finiteness*.

Kata kunci—Algoritma Gale – Shapley, jejaring sosial, pencarian kerja, dan Universitas Multimedia Nusantara.

I. LATAR BELAKANG

Saat ini, pencarian kerja di Universitas Multimedia Nusantara (UMN) untuk mahasiswa masih dilakukan secara langsung dengan cara bertanya jawab antara mahasiswa yang bersangkutan dengan Departemen Pengembangan Karir Universitas Multimedia Nusantara, di mana belum ada sistem yang diimplementasikan dengan tujuan mencari dan memasang lowongan kerja dengan pencari kerja yang begitu banyak sehingga seiring dengan bertambahnya jumlah mahasiswa dan alumni UMN, proses pemasangan atau pencocokan di atas sulit dilakukan terus – menerus karena akan memakan waktu dan sumber daya manusia yang tidak sedikit. Selain itu, dalam proses pemasangan di atas akan sulit menjamin apakah lowongan kerja yang disarankan sudah sesuai dengan bakat dan minat mahasiswa atau belum, begitu juga sebaliknya apakah mahasiswa yang mendaftar sudah sesuai dengan kualifikasi perusahaan atau belum karena kita tidak memiliki *preference list* dari kedua belah pihak yaitu mahasiswa yang bertindak sebagai pencari kerja dan perusahaan yang bertindak sebagai penyedia lowongan kerja. Lantas, bagaimana membuat suatu sistem yang dapat

memasangkan atau mencocokkan antara pencari kerja dalam hal ini *civitas academica* UMN dengan lowongan kerja yang diterbitkan oleh suatu institusi atau perusahaan berdasarkan daftar pilihan / keinginan mahasiswa dan perusahaan?

Pada 1962, David Gale dan Lloyd Shapley memperkenalkan studi pencocokan untuk membuat alokasi himpunan pasangan – pasangan stabil yang kemudian dikenal dengan *Stable Marriage Problem*. Penyelesaian *Stable Marriage Problem* bertujuan untuk mencari pasangan – pasangan yang stabil dari sejumlah n pria dan sejumlah n wanita yang memiliki urutan ketertarikan sendiri terhadap calon pasangan lainnya yang berbeda jenis. Mereka juga memberikan cara penyelesaian masalah di atas dengan memperkenalkan Algoritma Gale – Shapley. Algoritma Gale – Shapley telah dipakai oleh Alvin E. Roth untuk mencocokkan klien, seperti dalam kasus pelajar dengan sekolah, pendonor dengan pasien yang memerlukan transplantasi organ, dan dokter dengan rumah sakit. Kombinasi teori Shapley dan investigasi empiris Roth membawa mereka meraih Nobel pada 2012 di bidang sains ekonomi untuk teori alokasi stabil dan aplikasi dalam *market design* [1]. Berdasarkan pemaparan di atas, penelitian ini berfokus pada bagaimana mengimplementasikan Algoritma Gale – Shapley dalam Situs Jejaring Sosial Pencarian Kerja UMN Vacancy dengan tujuan memasang pencari kerja (*civitas academica* UMN) dengan lowongan kerja suatu institusi atau perusahaan dan menyediakan sarana publikasi lowongan kerja bagi perusahaan.

II. STABLE MARRIAGE PROBLEM

Pertama kali diperkenalkan oleh David Gale dan Lloyd Shapley dalam *paper* seminar mereka yang berjudul *College Admissions and the Stability of Marriage* pada 1962. Penyelesaian *Stable Marriage Problem* bertujuan untuk mencari pasangan – pasangan yang stabil dari sejumlah n pria dan sejumlah n wanita yang memiliki urutan ketertarikan sendiri terhadap calon pasangan lainnya yang berbeda jenis. Peneliti menyatakan bahwa untuk setiap jumlah pria dan wanita yang sama, selalu dimungkinkan untuk menyelesaikan *Stable Marriage Problem* dan membuat *matching* tersebut stabil. Untuk itu peneliti memperkenalkan algoritma penyelesaian masalah di atas yang dikenal dengan *Gale – Shapley Algorithm* [2].

III. GALE – SHAPLEY ALGORITHM

Tujuan utama dari algoritma ini adalah memasangkan sejumlah n pria dan n wanita dengan syarat monogami (satu pria untuk satu wanita, begitu pula sebaliknya) dan heteroseksual (antara pria dan wanita) berdasarkan *preference list* yang dibuat oleh pria dan wanita sehingga terbentuk himpunan M yang terdiri dari pasangan – pasangan yang stabil. *Preference list* di sini adalah daftar urutan pria dan wanita berdasarkan tingkat ketertarikan mulai dari yang paling diminati, yang kedua diminati apabila tidak cocok dengan orang pertama, yang ketiga diminati apabila tidak cocok dengan orang kedua, dan seterusnya hingga yang ke – n diminati apabila tidak cocok dengan orang yang ke – $(n-1)$.

Misalnya, sejumlah n pria kita notasikan dengan (A, B, C, \dots) dan sejumlah n wanita kita notasikan dengan (a, b, c, \dots) . Ketika kita memiliki pasangan $X-a$ dan $Y-b$, jika X lebih menyukai b dibandingkan dengan pasangannya saat ini, yaitu a dan b lebih menyukai X dibandingkan dengan pasangannya saat ini, yaitu Y , maka $X-b$ disebut pasangan yang tidak stabil (*dissatisfied pair*). Himpunan M dikatakan stabil apabila tidak memiliki pasangan yang tidak stabil (*dissatisfied pairs*).

```

Function PROPOSAL-ALGORITHM
1: while there is an unpaired man do
2:   pick an unpaired man  $X$  and the first woman  $w$  on his list
3:   remove  $w$  from his list so it won't be picked again
4:   if  $w$  is engaged then
5:     if  $w$  prefers  $X$  more than her current partner  $Y$  then
6:       set  $X-w$  as married
7:       set  $Y-w$  as unmarried so now  $Y$  is unpaired
8:     else
9:        $X$  is still unpaired since  $w$  is happier with  $Y$ 
10:    end if
11:   else
12:     the woman was not previously paired so accept immediately,  $X-w$ , as married
13:   end if
14: end while

```

Gambar 1. Proposal Algorithm

Dalam perhitungan statistika jika kita memiliki n pria dan n wanita, maka paling banyak proposal yang diajukan untuk mendapatkan himpunan M yang terdiri dari pasangan – pasangan stabil adalah n^2 [3].

Untuk lebih jelasnya, kita mulai dari *preference list* berikut ini [4].

TABEL I. WOMEN PREFERENCES

Women	Men				
	Joe	Brian	George	Matt	Jim
Amy	1	2	4	3	5
Sarah	3	5	1	2	4
Susan	5	4	2	1	3
Kelly	1	3	5	4	2
Dianne	4	2	3	5	1

TABEL II. MEN PREFERENCES

Men	Women				
	Amy	Sarah	Susan	Kelly	Dianne
Joe	5	1	2	4	3
Brian	4	1	3	2	5
George	5	3	2	4	1
Matt	1	5	4	3	2
Jim	4	3	2	1	5

Setiap pria akan melamar wanita yang menjadi prioritas utamanya, sedangkan setiap wanita akan mengikuti aturan berikut:

1. Jika seorang wanita belum bertunangan dan belum dilamar, maka ia harus menunggu.
2. Jika seorang wanita belum bertunangan, tetapi sedang dilamar, maka ia akan menerima lamaran tersebut.
3. Jika seorang wanita belum bertunangan, tetapi telah memiliki banyak lamaran (lebih dari satu), maka ia akan menerima lamaran dari pria yang menduduki *preference list* tertinggi.
4. Jika seorang wanita telah bertunangan dan menerima lamaran lain, maka ia akan menerima lamaran dari pria yang menduduki *preference list* tertinggi.

Putaran I:

Men : Setiap pria akan melamar wanita yang menduduki *preference list* tertinggi.

Women : Setiap wanita akan mengikuti empat aturan di atas.

Men: Joe proposes to Sarah	Women: Sarah accepts Joe's proposal
Brian proposes to Sarah	Sarah does not take Brian's proposal
George proposes to Dianne	Dianne accepts George's proposal
Matt proposes to Amy	Amy accepts Matt's proposal
Jim proposes to Kelly	Kelly accepts Jim's proposal

Gambar 2. Putaran Pertama Algoritma Gale – Shapley

Putaran II:

Men : Setiap pria yang belum bertunangan akan melamar wanita yang merupakan *preference list* selanjutnya.

Women : Setiap wanita akan mengikuti empat aturan di atas.

Men: Brian proposes to Kelly	Women: Kelly doesn't change (Jim is higher)
------------------------------	---

Gambar 3. Putaran Kedua Algoritma Gale – Shapley

Putaran III:

Men : Setiap pria yang belum bertunangan akan melamar wanita yang merupakan *preference list* selanjutnya.

Women : Setiap wanita akan mengikuti empat aturan di atas.

Men: Brian proposes to Susan	Women: Susan accepts Brian's proposal
Now we have five stable couples.	
Joe	Sarah
Brian	Susan
George	Dianne
Matt	Amy
Jim	Kelly

Gambar 4. Putaran Ketiga Algoritma Gale – Shapley

Dalam kenyataannya, kita dihadapkan pada kondisi di mana pencocokan tidak hanya terjadi pada *one to one*, tetapi juga *one to many* seperti pada kasus *Hospitals Residents*, *College Admission Problem*, di mana satu atau lebih rumah sakit bisa menampung satu atau lebih dokter, satu atau lebih jurusan bisa menampung satu atau lebih mahasiswa. Selain itu, dalam kasus pencocokan pria dan wanita pada Algoritma Gale – Shapley, setiap pria dan wanita harus menetapkan urutan ketertarikan terhadap pasangan lain yang berbeda jenis dengan asumsi bahwa setiap pria dan wanita akan bahagia bila dicocokkan dengan pria dan wanita lain yang kurang disukai daripada tidak mendapatkan pasangan sama sekali. Dalam perkembangannya, Algoritma Gale – Shapley diperbarui sehingga kedua masalah di atas dapat diatasi dan penerapannya lebih sesuai dengan keadaan pencocokan di dunia nyata [5].

IV. HASIL UJI COBA

A. Evaluasi Algoritma Gale – Shapley

People dalam Algoritma Gale – Shapley merupakan pihak yang aktif melakukan *propose* (pengajuan lamaran kerja) berdasarkan *preference list* mereka sehingga alur algoritma ini dapat dianalogikan sebagai seorang pencari kerja yang melamar di lowongan perusahaan tertentu, di mana pada dasarnya pencari kerja akan mendatangi pihak yang menjadi *preference list* utamanya, begitu seterusnya hingga ia diterima di perusahaan.

Vacancy dalam Algoritma Gale – Shapley merupakan pihak yang pasif (tidak melakukan *propose*), tetapi bertugas untuk menentukan pencari kerja yang layak untuk mengisi posisi lowongan berdasarkan pilihan yang terdapat pada *preference list Vacancy* sehingga *proposal* akan diterima apabila posisi lowongan yang tersedia masih ada dan *People* yang melamar memiliki *ranking* terbaik dari *preference list Vacancy*.

Dalam proses *pairing* antara *People* dan *Vacancy* dapat terjadi beberapa kasus, di mana terjadi *engagement* dan *unengagement* berdasarkan kondisi berikut ini:

1. Jika *People* merupakan pendaftar pertama, tetapi *Vacancy* tidak memilih *People* yang bersangkutan

dalam *preference list* mereka, maka tidak terjadi *engagement* atau *unengagement*.

2. Jika *People* merupakan pendaftar pertama, *Vacancy* masih tersedia dan telah memilih *People* yang bersangkutan dalam *preference list*, maka terjadi *engagement*.
3. *People* bukan merupakan pendaftar pertama, *Vacancy* sudah terisi dengan *People* yang mendaftar sebelumnya dan telah memilih *People* yang bersangkutan dalam *preference list*.
 - a. Jika *ranking People* saat ini lebih tinggi dari *ranking fiance* dari *Vacancy* yang berada di peringkat terendah, maka terjadi *unengagement* untuk *fiance* dan *engagement* untuk *People* saat ini.
 - b. Jika *ranking People* saat ini lebih rendah dari *ranking fiance* dari *Vacancy* yang berada di peringkat terendah, maka tidak terjadi *engagement* atau *unengagement*.

Jika kita melihat bagaimana *pairing* dilakukan baik untuk *People* saat ini, maupun *People* yang sudah diterima sebelumnya, maka proses *engagement* untuk kandidat yang menang dan *unengagement* untuk kandidat yang kalah akan selalu menjamin bahwa *pairing* yang dilakukan adalah yang paling stabil karena proses *engagement* atau *unengagement* tersebut berdasarkan pertimbangan urutan *preference list* yang akan dilepas atau diterima terkait *Vacancy* tertentu.

Begitu pula jika kita menilik pencocokan yang stabil dari sisi *People* sendiri berdasarkan pencocokan yang dimulai dari *preference list* tertinggi hingga *preference list* terendah, maka dalam tingkatan tertentu ketika *People* telah dipilih oleh *Vacancy*, pencocokan terhadap *People* dengan *preference list* tertinggi akan berhenti selama tidak terjadi *unengagement* karena kalah dalam *ranking*.

B. Skenario Uji Coba

Berikut merupakan contoh skenario yang menunjukkan bagaimana Algoritma Gale – Shapley berjalan dan kebenarannya:

1. Skenario pertama pengujian parameter *input*, *output*, *finiteness*, dan *definiteness*. *Pairing* akan dimulai dari Ivan Bong.

TABEL III. PEOPLE SKENARIO PERTAMA

Urutan	People Preference List	
	People	Preference
1	Ivan Bong	Web Developer, Database Admin
2	Ivan Arya Putra	Database Admin, Android Programmer, Web Developer
3	Tampan Wibawa Guna	Web Developer, Database Admin
4	Engelberta Bong	Web Developer

TABEL IV. VACANCY SKENARIO PERTAMA

Nomor	Vacancy Preference List		
	Vacancy	Preference	Position
1	Web Developer	Ivan Bong, Tampan Wibawa Guna	1
2	Database Admin	Tampan Wibawa Guna, Ivan Arya Putra	1
3	Android Programmer	Tidak ada	3

Pairing dimulai dari Ivan Bong yang menunjuk Web Developer sebagai pilihan utamanya. Web Developer memilih Ivan Bong di urutan teratas dan posisi lowongan kerja masih tersedia satu. Oleh karena itu, Ivan Bong akan langsung dipasangkan dengan Web Developer.

Selanjutnya, Ivan Arya Putra menunjuk Database Admin sebagai pilihan utamanya. Database Admin memilih Ivan Arya Putra di urutan kedua dan posisi lowongan kerja masih tersedia satu. Oleh karena itu, Ivan Arya Putra akan langsung dipasangkan dengan Database Admin.

Tampan Wibawa Guna menunjuk Web Developer sebagai pilihan utamanya. Web Developer memilih Tampan Wibawa Guna di urutan kedua, tetapi posisi lowongan kerja sudah diisi. *Ranking* Tampan Wibawa Guna dibandingkan dengan *ranking* Ivan Bong, di mana Ivan Bong tidak digantikan karena memiliki *ranking* lebih tinggi.

Tampan Wibawa Guna menunjuk Database Admin sebagai pilihan kedua setelah ditolak di Web Developer. Database Admin memilih Tampan Wibawa Guna di urutan teratas, tetapi posisi lowongan kerja sudah diisi oleh Ivan Arya Putra. *Ranking* Tampan Wibawa Guna dibandingkan dengan *ranking* Ivan Arya Putra, di mana Ivan Arya Putra akan digantikan karena memiliki *ranking* lebih rendah dan Tampan Wibawa Guna dipasangkan dengan Database Admin.

Sementara itu, Ivan Arya Putra menunjuk Android Programmer sebagai pilihan kedua setelah ditolak di Database Admin. Android Programmer tidak membuat *preference list* walaupun lowongan masih tersedia tiga tempat.

Oleh karena itu, Ivan Arya menunjuk Web Developer sebagai opsi terakhirnya. Web Developer tidak menunjuk Ivan Arya Putra dan pencocokan untuk Ivan Arya Putra berhenti karena *preference list* yang ia buat telah habis sehingga ia tidak mendapatkan pekerjaan.

Pencocokan dilanjutkan dengan Engelberta Bong yang memilih Web Developer sebagai pilihan utama. Web Developer tidak memilih Engelberta Bong dan *preference list* Engelberta Bong juga telah habis, sehingga baik Ivan Arya Putra dan Engelberta Bong tidak mendapatkan pekerjaan yang sesuai.

Dari hasil ujicoba di atas dapat disimpulkan bahwa program berjalan dengan *state* yang *finiteness*, memiliki *input People*, *Vacancy*, *People preList*, *Vacancy*

preList, dan menghasilkan *output* status pencocokan untuk setiap *People* dan *Vacancy*.

2. Skenario kedua pengujian parameter *input*, *output*, *finiteness*, dan *definiteness*. *Pairing* akan dimulai dari Ivan Arya Putra.

TABEL V. PEOPLE SKENARIO KEDUA

Urutan	People Preference List	
	People	Preference
1	Ivan Arya Putra	Database Admin, Android Programmer, Web Developer
2	Ivan Bong	Web Developer, Database Admin
3	Engelberta Bong	Web Developer
4	Tampan Wibawa Guna	Web Developer, Database Admin

TABEL VI. VACANCY SKENARIO KEDUA

Nomor	Vacancy Preference List		
	Vacancy	Preference	Position
1	Web Developer	Ivan Bong, Tampan Wibawa Guna	1
2	Database Admin	Tampan Wibawa Guna, Ivan Arya Putra	1
3	Android Programmer	Tidak ada	3

Ivan Arya Putra memilih Database Admin di urutan pertama dan Database Admin memilih Ivan Arya Putra di urutan kedua (posisi masih tersedia satu), Ivan Arya Putra kemudian dipasangkan dengan Database Admin.

Ivan Bong memilih Web Developer di urutan teratas dan Web Developer juga demikian halnya memilih Ivan Bong di urutan teratas (posisi masih tersedia satu), Ivan Bong kemudian dipasangkan dengan Web Developer.

Engelberta Bong memilih Web Developer di urutan pertama, Web Developer sudah terisi penuh dan tidak memilih Engelberta Bong. Oleh karena itu, Engelberta Bong tidak mendapatkan pekerjaan karena tidak ada *preference list* lagi.

Tampan Wibawa Guna memilih Web Developer di urutan teratas. Web Developer juga memilih Tampan Wibawa Guna di posisi kedua. *Ranking* Tampan Wibawa Guna dibandingkan dengan *ranking* Ivan Bong. Ivan Bong tidak digantikan karena Web Developer memilih Ivan Bong di urutan yang lebih tinggi daripada Tampan Wibawa Guna.

Tampan Wibawa Guna beralih ke Database Admin. Database Admin sudah terisi oleh Ivan Arya Putra sehingga *ranking* Tampan Wibawa Guna kemudian dibandingkan dengan *ranking* Ivan Arya Putra. Ivan Arya Putra dilepas dari Database Admin karena kalah *ranking* dengan Tampan Wibawa Guna.

Oleh karena itu, Ivan Arya Putra kemudian beralih ke Android Programmer. Android Programmer tidak memilih seseorang dalam *preference list*.

Ivan Arya Putra beralih ke Web Developer. Web Developer sudah penuh karena terisi oleh Ivan Bong. *Ranking* Ivan Arya Putra kemudian dibandingkan dengan *ranking* Ivan Bong. Ivan Arya Putra tidak mendapatkan pekerjaan Web Developer karena memiliki *ranking* lebih rendah daripada Ivan Bong. Ivan Arya Putra tidak memiliki *preference list* lagi sehingga pencocokan berhenti.

Dari hasil ujicoba di atas dapat disimpulkan bahwa program berjalan dengan *state* yang *finiteness*, memiliki *input People*, *Vacancy*, *People preList*, *Vacancy preList*, dan menghasilkan *output* status pencocokan untuk setiap *People* dan *Vacancy*.

3. Berdasarkan hasil pengujian kedua skenario di atas, di mana masing – masing skenario menghasilkan *output* status yang sama untuk *People* dan *Vacancy*, dapat disimpulkan bahwa Algoritma Gale – Shapley memenuhi kriteria *definiteness*.

C. Implementasi pada Subdomain UMN

Situs Jejaring Sosial Pencarian Kerja UMN Vacancy telah diimplementasikan pada *subdomain* Universitas Multimedia Nusantara yang dapat diakses dengan menggunakan alamat <http://cdc.umn.ac.id> dan namanya diubah menjadi Career Development Center. Career Development Center akan memfasilitasi mahasiswa atau alumni UMN yang ingin mencari tempat kerja magang atau tempat kerja tetap, perusahaan yang ingin melakukan publikasi lowongan kerja dan mencari karyawan, dan Departemen Pengembangan Karir dan Alumni UMN yang ingin mengetahui mahasiswa atau lulusan yang telah melamar dan bekerja di suatu perusahaan tertentu.

V. SIMPULAN

Simpulan yang diperoleh setelah melakukan penelitian ini adalah

1. Algoritma Gale – Shapley berhasil mencocokkan pencari kerja dengan lowongan yang ada sehingga terbentuk himpunan pasangan stabil pada Situs UMN Vacancy.
2. Dalam skenario yang sudah diuji dapat dilihat bahwa tidak peduli siapa yang terlebih dahulu dipasangkan, Algoritma Gale – Shapley menghasilkan *output*

pasangan yang sama walaupun dengan urutan pencocokan yang berbeda – beda sehingga dikatakan stabil, konsisten, dan memenuhi aturan algoritma *definiteness*.

3. Dalam skenario yang sudah diuji dapat dilihat bahwa Algoritma Gale – Shapley selalu memiliki *state* yang *finiteness* hingga menghasilkan *output*.
4. Syarat untuk menjalankan Algoritma Gale – Shapley adalah adanya ketertarikan antara kedua pihak, baik *People*, maupun *Vacancy* sehingga jika *People* membuat *preference list* yang berisi daftar *Vacancy* pilihannya dan *Vacancy* juga membuat *preference list* yang berisi daftar *People* pilihannya, maka setiap pemasangan yang dilakukan antara *People* dan *Vacancy* dalam satu himpunan dapat dikatakan sebagai alokasi pasangan – pasangan yang stabil. Namun, bukan berarti bahwa semua *preference list* adalah calon pasangan stabil, melainkan jika tidak ada pasangan lain lagi yang paling sesuai dengan pasangan sekarang, maka pasangan sekarang adalah pasangan yang paling ideal.
5. Dalam suatu kondisi bisa saja terjadi *unengagement* karena peringkat agen tertentu yang lebih rendah daripada agen lainnya sehingga tidak mendapatkan pasangan. Untuk itu, semakin banyak *preference list* yang dibuat oleh suatu agen, semakin besar pula kemungkinan agen tersebut mendapatkan pasangan. Begitu pula sebaliknya, semakin sedikit *preference list* yang dibuat oleh suatu agen, semakin kecil pula kemungkinan agen tersebut mendapatkan pasangan, kecuali agen tersebut menempati urutan pertama (*top list*) dalam *preference list* agen lain yang akan dicocokkan dengannya.

DAFTAR PUSTAKA

- [1] The Royal Swedish Academy of Sciences. (2012). The Prize in Economic Sciences 2012. Stockholm.
- [2] G. Iwama, K., & Miyazaki, S. (2008). A Survey of the Stable Marriage Problem and Its Variants. Kyoto University.
- [3] Hunt, W. The Stable Marriage Problem. West Virginia University.
- [4] Premer, J. (2011). Stable Marriage Problem.
- [5] Shorey, R. Hospitals and Residents. Swarthmore College. <http://www.sccs.swarthmore.edu/users/06/rshorey/math/hospitals.html>.