

LAPORAN PRAKTIKUM ANALISIS ALGORITMA



Disusun Oleh :

Rio Sapta Samudera -140810180030

**Program Studi S-1 Teknik Informatika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran
2020**

1. Landasan Teori

Stable Matching Problem (SMP) adalah problem algoritmik yang memberikan ilustrasi mengenai berbagai tema yang dipelajari di analisis algoritma ini. Algoritma ini muncul dari beberapa problem

praktis. Oleh karena itu supaya problemnya jelas dan penyelesaian tepat perlu dilakukan 3 langkah

berikut:

- Mencermati problem
- Memformulasikan problem
- Mendesain algoritma

Stable Matching Problem berasal, sebagian, pada tahun 1962, ketika David Gale dan Lloyd Shapley, dua matematika ekonom, mengajukan pertanyaan:

Bisakah seseorang merancang sebuah perguruan tinggi proses penerimaan, atau proses perekrutan pekerjaan, itu mandiri (otomatis)?

Inti dari proses aplikasi adalah interaksi antara dua jenis pihak yang berbeda: perusahaan dan pelamar.

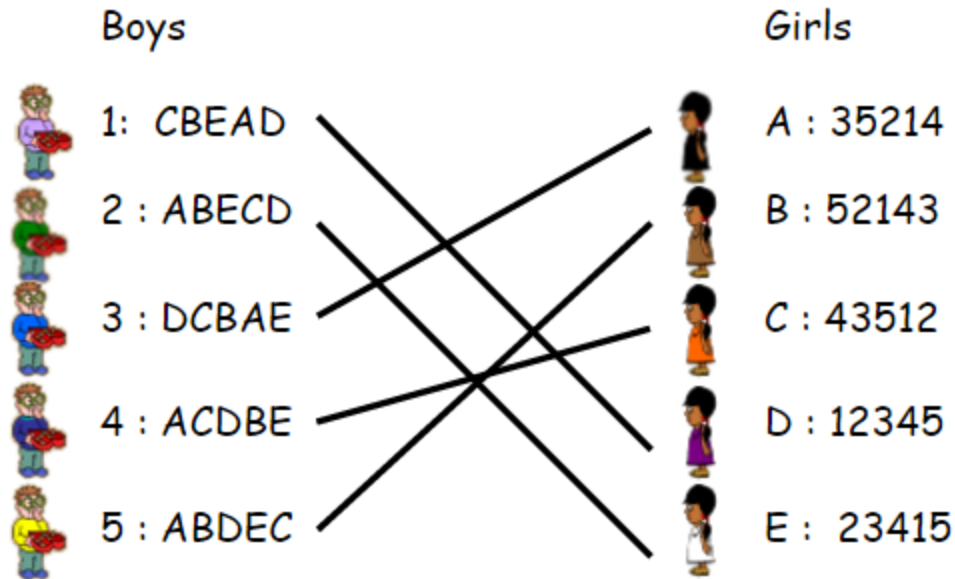
Setiap pelamar memiliki daftar preferensi perusahaan yang ingin dimasuki, dan setiap perusahaan-setelah aplikasi masuk-membentuk daftar preferensi akan pelamarnya. Berdasarkan preferensi ini, perusahaan memberikan penawaran kepada beberapa pelamar mereka, pelamar memilih penawaran mana yang akan mereka terima.

Bagaimana jika tidak dilakukan secara otomatis? Kemungkinan resiko kecurangan tinggi.

Jadi inilah pertanyaan yang diajukan Gale dan Shapley: Diberikan seperangkat preferensi di antara pemberi kerja dan pelamar, dapatkah kami menetapkan pelamar untuk pemberi kerja sehingga untuk setiap pemberi kerja E , dan setiap pelamar A yang tidak dijadwalkan bekerja untuk E , setidaknya satu dari dua hal berikut ini yang terjadi?

- (i) E lebih memilih setiap satu dari daftar pelamar yang diterima(A); atau
- (ii) A lebih suka situasinya saat ini daripada bekerja untuk pemberi kerja E

Jika ini berlaku, hasilnya stabil: kepentingan pribadi individu akan mencegah kesepakatan pemohon/pemberi kerja dibuat dibalik layar. Gale dan Shapley mengembangkan solusi algoritmik yang tajam untuk problem ini, yang akan kita pelajari.



Berdasarkan daftar preferensi ini, pria akan melamar wanita dan wanita memilih pria mana yang akan mereka terima. Bagaimana caranya agar bisa dilakukan secara otomatis?

Pseudocode Algoritma Gale Shapley

untuk setiap pria M yang belum berpasangan:

W <- wanita teratas dalam preferensi M yang belum dilamar M

jika W sudah dilamar:

jika preferensi W terhadap M > pasangan(W):

W dan M berpasangan

lainnya:

M ditolak oleh W

lainnya:

W dan M berpasangan

Tabel Preferensi Pria dan Wanita

1	2	4	1	3	1	2	1	4	3
2	3	1	4	2	2	4	3	1	2
3	2	3	1	4	3	1	4	3	2
4	4	1	3	2	4	2	1	4	3
Men's Preferences					Women's Preferences				

Dengan Algoritma Gale Shapley didapat pasangan (Pria, Wanita) sebagai berikut :

{ (1,4), (2,3), (3,2), (4,1) }

Ada dua pasangan (m, w) dan (m', w') dengan kondisi :

m lebih suka w' daripada w ,

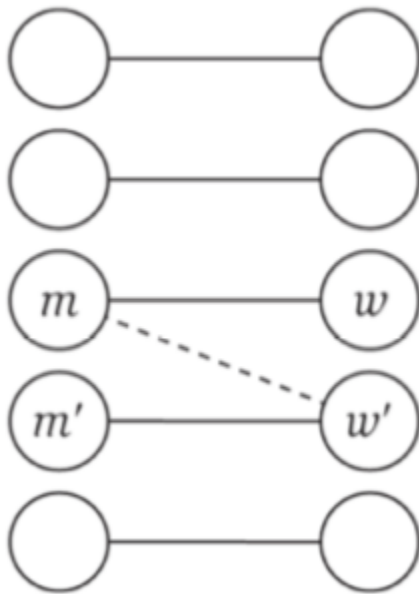
w' lebih suka ke m' .

Dalam hal ini, tidak ada yang bisa menghentikan m dan w' untuk meninggalkan pasangan mereka saat ini dan pergi bersama.

Set pernikahan menjadi tidak self-enforcing.

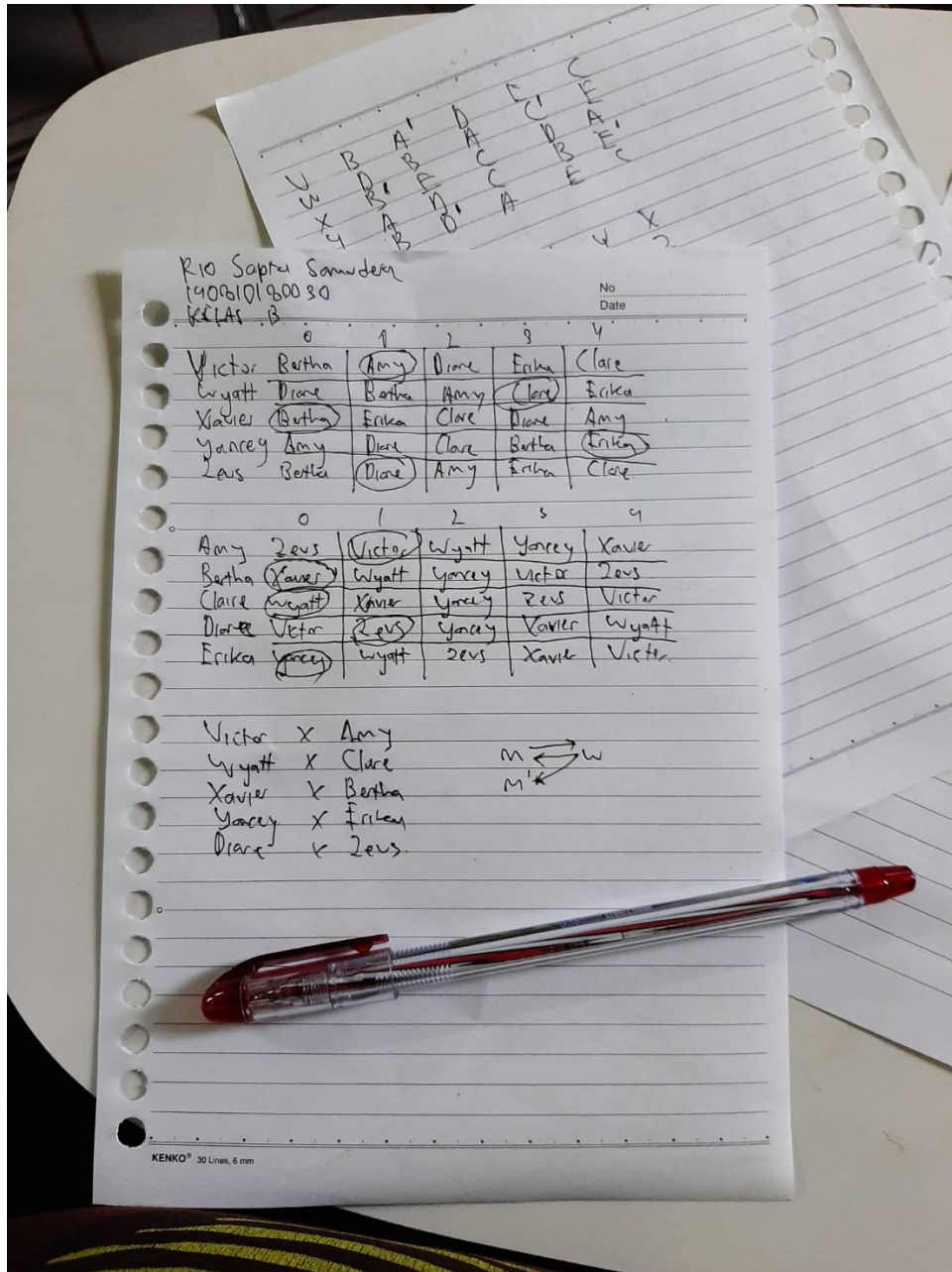
Jadi pasangan (m, w) dan (m', w') tidak stabil

Stabilnya gimana? $\rightarrow (m, w')$ dan $(m', w) \rightarrow$ self enforcing



2. Tugas

1. Worksheet 1



2. Program C++

```

1. #include <iostream>
2. #include <string.h>
3. #include <stdio.h>
4. using namespace std;
5.
6. #define N 5
7.
8. bool wPrefersM1OverM(int prefer[2 * N][N], int w, int m, int m1)
9. {

```

```

10.     for (int i = 0; i < N; i++)
11.     {
12.         if (prefer[w][i] == m1)
13.             return true;
14.         if (prefer[w][i] == m)
15.             return false;
16.     }
17. }
18.
19. void stableMarriage(int prefer[2 * N][N], string name[10])
20. {
21.
22.     int wPartner[N];
23.     bool mFree[N];
24.
25.     memset(wPartner, -1, sizeof(wPartner));
26.     memset(mFree, false, sizeof(mFree));
27.     int freeCount = N;
28.
29.     while (freeCount > 0)
30.     {
31.         int m;
32.         for (m = 0; m < N; m++)
33.             if (mFree[m] == false)
34.                 break;
35.         for (int i = 0; i < N && mFree[m] == false; i++)
36.         {
37.             int w = prefer[m][i];
38.             if (wPartner[w - N] == -1)
39.             {
40.                 wPartner[w - N] = m;
41.                 mFree[m] = true;
42.                 freeCount--;
43.             }
44.             else
45.             {
46.                 int m1 = wPartner[w - N];
47.                 if (wPrefersM1OverM(prefer, w, m, m1) == false)
48.                 {
49.                     wPartner[w - N] = m;
50.                     mFree[m] = true;
51.                     mFree[m1] = false;
52.                 }
53.             }
54.         }
55.     }
56.     cout << "Woman Man" << endl;
57.     for (int i = 0; i < N; i++)
58.         cout << " " << name[i + N] << "\t" << name[wPartner[i]] << endl;
59. }
60.
61. int main()
62. {
63.     int prefer[2 * N][N] = { {6, 5, 8, 9, 7},
64.                               {8, 6, 5, 7, 9},
65.                               {6, 9, 7, 8, 5},
66.                               {5, 8, 7, 6, 9},
67.                               {6, 8, 5, 9, 7},

```

```

68.         {4, 0, 1, 3, 3},
69.         {2, 1, 3, 0, 4},
70.         {1, 2, 3, 4, 0},
71.         {0, 4, 3, 2, 1},
72.         {3, 1, 4, 2, 0},
73.     };
74.     string name[10] = { "Victor", "Wyatt", "Xavier", "Yancey", "Zeus", "Amy",
    "Bertha", "Clare", "Diane", "Erika" };
75.     stableMarriage(prefer, name);
76.
77.     return 0;
78. }

```

3. Analisis Algoritma

1. Apakah jawaban Anda di Worksheet 01 dan Program sama persis? Jika Tidak? Kenapa?

Ya, sama persis.

Anda diminta untuk membuktikan algoritma G-S benar dengan menjawab pertanyaan berikut:

Fakta (1.1):

Seorang wanita tetap bertunangan dari titik di mana dia menerima proposal pertamanya; dan urutan mitra yang bertunangan dengannya menjadi lebih baik dan lebih baik lagi (hal ini sesuai dengan daftar preferensi wanita). -> tidak perlu dipertanyakan

Fakta (1.2):

Urutan wanita yang dilamar pria lebih buruk dan lebih buruk lagi (hal ini sesuai dengan daftar preferensi pria). -> tidak perlu dipertanyakan

Teorema (1.3):

Algoritma G-S berakhir setelah paling banyak n^2 iterasi menggunakan While Loop. Buktikan!

Jumlah wanita dan pria sama-sama berjumlah n , maka jika pria melamar wanita dan tidak diterima, maka pengulangannya hanya sebanyak wanita dan pria yang ada, sehingga maksimal iterasi adalah n^2

Teorema (1.4):

Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan. Buktikan!

Karena ada jumlah yang sama antara pria dan wanita dan tidak ada yang bisa menikah ke lebih dari satu orang. Dan tidak mungkin algoritma tersebut selesai karena m harus menikah, kepada w , yang mana dia tidak melamarnya, atau dia ditolak oleh semua wanita, termasuk w . Hingga pada akhirnya m dan w menikah, baru program berhenti.

Teorema (1.5):

Himpunan S yang dikembalikan saat terminasi adalah perfect matching Buktikan!

Asumsikan S bukan perfect matching.

Karena S bukan perfect matching, maka akan ada pria, m , yang masih bebas pada akhir algoritma. Mirip dengan kasus teorema 1.4, karena algoritmanya selalu memilih pria bebas untuk melamar ke setiap wanita yang belum dia lamar, maka sang pria sudah melamar kesemua wanita tapi masih tetap bebas. Maka dari itu bisa disimpulkan bahwa m ditolak atau ditinggalkan oleh setiap w untuk menikah dengan m' . Artinya setiap w sudah menikah dan mereka tetap menikah. Jadi, ada n wanita yang menikah dengan $n - 1$ pria (karena m bebas) yang berarti kontradiksi karena setidaknya ada pria yang menikah dengan multiple woman yang mana tidak bisa terjadi dalam algoritma ini.

Teorema (1.6):

Sebuah eksekusi algoritma G-S mengembalikan satu set pasangan S . Set S adalah pasangan yang stabil.

Buktikan!

Asumsikan jika S bukan pasangan yang stabil.

Maka akan ada pasangan, m, w' , yang tidak dipasangkan oleh algoritmanya, sehingga m lebih memilih w' daripada w , pasangannya, dan w' lebih memilih m daripada m' , pasangannya. Maka m melamar kepada w' sebelum melamar ke w , karena w' berada diatas w pada daftar preferensinya. Jadi jika seorang wanita menolak pria, maka dia lebih memilih pasangan akhirnya daripada pria yang ditolaknya. Maka w' lebih memilih m' daripada m , yang mana kontradiksi. Jadi Algoritma G-S memproduksi pasangan yang stabil.

Karena Algoritma Gale-Shapley selalu berakhir dan hasilnya adalah pasangan yang stabil, maka akan ada pernikahan yang stabil untuk setiap problema stable marriage.

3. Kesimpulan

Algoritma Gale-Shapley menghasilkan pasangan-pasangan yang stabil dan perfect-matching. Karena, terdapat kontradiksi apabila pasangan tersebut tidak stabil dan perfect-matching. Sehingga dapat diaplikasikan kepada *Stable Marriage Problem*.