LAPORAN PRAKTIKUM

ANALISIS ALGORITMA

Disusun oleh:

Fadlan Mulya Priatna

140810180041

Kelas A

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

1. **Materi**

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 preferensi pemesanan pada perusahaan, dan setiap perusahaan — setelah aplikasi masuk — membentuk pemesanan preferensi pada pelamarnya. Berdasarkan preferensi ini, perusahaan memberikan penawaran kepada beberapa pelamar mereka, pelamar memilih penawaran mana yang akan mereka terima.

Gale dan Shapley mempertimbangkan hal-hal yang dapat mulai salah dengan proses ini, tanpa adanya mekanisme untuk menegakkan status quo.

Banyak orang — baik pelamar maupun pemberi kerja — bias berakhir tidak senang dengan proses maupun hasilnya. Apa yang salah?

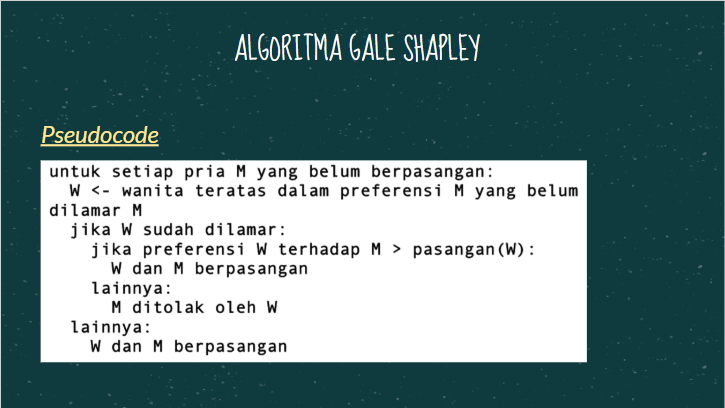
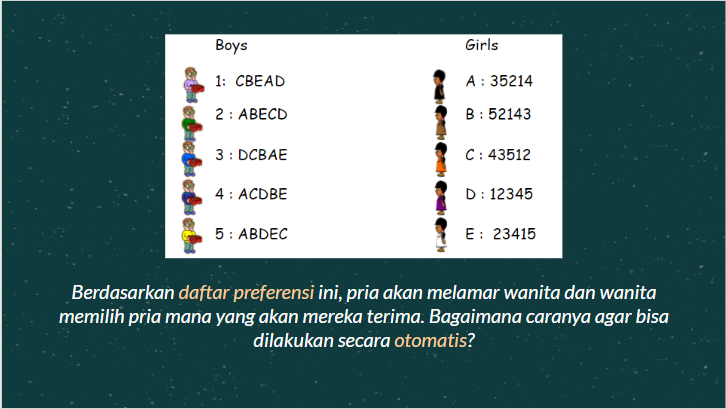
Satu masalah mendasar adalah bahwa prosesnya tidak self-enforcing (otomatis) — jika orang diizinkan untuk bertindak demi kepentingan diri mereka sendiri, maka proses itu berisiko hancur.

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?

1. E lebih memilih setiap satu dari daftar pelamar yang diterima (A); atau
2. A lebih suka situasinya saat ini dari pada bekerja untuk pemberi kerja E

Jika ini berlaku, hasilnya stabil: kepentingan pribadi individu akan mencegah kesepakatan pemohon/ pemberi kerja dibuat di balik layar.

Gale dan Shapley mengembangkan solusi algoritmik yang tajam untuk masalah ini, yang akan kita pelajari.



1. **Worksheet 1**

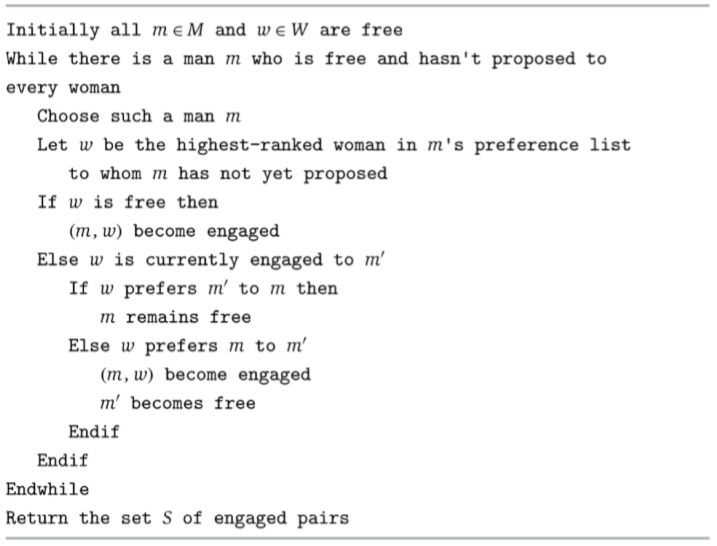
Dengan Algoritma Gale-Shapley, cari himpunan stable-matching yang sesuai dengan preferencelists berikut ini. Gunakan processor terhebat yang Anda miliki (otak) untuk mengikuti algoritma GS dan uraikan outputnya untuk setiap loop hingga menghasilkan stable-matching.

1. Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yancey | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

1. Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |



Iterasi 1

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yance | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Women’s

Men’s

Iterasi 2

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yance | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Women’s

Men’s

Iterasi 3

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yance | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Women’s

Men’s

Iterasi 4

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yance | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Women’s

Men’s

Iterasi 5

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yancey | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Men’s

Women’s

Iterasi 6

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yancey | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Men’s

Women’s

Iterasi 7

Men’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yancey | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

Women’s Preferences Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| Amy | Zeus | Victor | Wyatt | Yance | Xavier |
| Bertha | Xavier | Wyatt | Yance | Victor | Zeus |
| Clare | Wyatt | Xavier | Yance | Zeus | Victor |
| Diane | Victor | Zeus | Yance | Xavier | Wyatt |
| Erika | Yance | Wyatt | Zeus | Xavier | Victor |

Men’s

Women’s

1. **Program C++**
   1. **Coding**

/\*

Nama Program : Stable Matching Problem

Nama : Fadlan Mulya Priatna

NPM : 140810180041

Tanggal Pembuatan : 2 Maret 2020

Deskripsi Program : Berisikan coding program pencari jodoh

\*/

#include<iostream>

using namespace std;

typedef int prefer[5];

struct orang{

string nama;

int pasangan;

int preferences[5];

};

orang setDataOrang(string nama, prefer prefers){

orang org;

org.nama = nama;

org.pasangan = -1;

for(int i = 0; i < 5;i++){

org.preferences[i] = prefers[i];

}

return org;

}

void stableMatching(orang pria[5], orang wanita[5]){

bool bebas = true;

int i=0, j=0, prefSek,prefSuk;

while(bebas){

if(wanita[pria[i].preferences[j]].pasangan = -1){

pria[i].pasangan = pria[i].preferences[j];

wanita[pria[i].preferences[j]].pasangan = i;

}

else{

int k = 0;

prefSek = -1; prefSuk = -1;

while(prefSek == -1 || prefSuk == -1){

if(i == wanita[pria[i].preferences[j]].preferences[k]){

prefSuk = k;

}

if(wanita[pria[i].preferences[j]].pasangan == wanita[pria[i].preferences[j]].preferences[k]){

prefSek = k;

}

k++;

}

if(prefSuk<prefSek){

pria[wanita[pria[i].preferences[j]].pasangan].pasangan = -1;

pria[i].pasangan = pria[i].preferences[j];

wanita[pria[i].preferences[j]].pasangan = i;

}

else{

j++;

continue;

}

}

bebas = false;

j=0;

for(int l = 0; l < 5; l++){

if(pria[l].pasangan == -1){

i = l;

bebas = true;

break;

}

}

}

}

int main(){

orang pria[5], wanita[5];

prefer prefers;

//Pria

prefers[0] = 1; prefers[1] = 0; prefers[2] = 3; prefers[3] = 4; prefers[4] = 2;

pria[0] = setDataOrang("Victor",prefers);

prefers[0] = 3; prefers[1] = 1; prefers[2] = 0; prefers[3] = 2; prefers[4] = 4;

pria[1] = setDataOrang("Wyatt",prefers);

prefers[0] = 1; prefers[1] = 4; prefers[2] = 2; prefers[3] = 3; prefers[4] = 0;

pria[2] = setDataOrang("Xavier",prefers);

prefers[0] = 0; prefers[1] = 3; prefers[2] = 2; prefers[3] = 1; prefers[4] = 4;

pria[3] = setDataOrang("Yancey",prefers);

prefers[0] = 1; prefers[1] = 3; prefers[2] = 0; prefers[3] = 4; prefers[4] = 2;

pria[4] = setDataOrang("Zeus",prefers);

//Wanita

prefers[0] = 4; prefers[1] = 0; prefers[2] = 1; prefers[3] = 3; prefers[4] = 2;

wanita[0] = setDataOrang("Amy",prefers);

prefers[0] = 2; prefers[1] = 1; prefers[2] = 3; prefers[3] = 0; prefers[4] = 4;

wanita[1] = setDataOrang("Bertha",prefers);

prefers[0] = 1; prefers[1] = 2; prefers[2] = 3; prefers[3] = 4; prefers[4] = 0;

wanita[2] = setDataOrang("Clare",prefers);

prefers[0] = 0; prefers[1] = 4; prefers[2] = 3; prefers[3] = 2; prefers[4] = 1;

wanita[3] = setDataOrang("Diane",prefers);

prefers[0] = 3; prefers[1] = 1; prefers[2] = 4; prefers[3] = 2; prefers[4] = 0;

wanita[4] = setDataOrang("Erika",prefers);

stableMatching(pria,wanita);

cout << "\nHasil Stable Matching :" << endl;

cout << "-----------------" << endl;

cout << "| Man \t Woman\t|" << endl;

cout << "-----------------" << endl;

for(int i = 0; i < 5; i++){

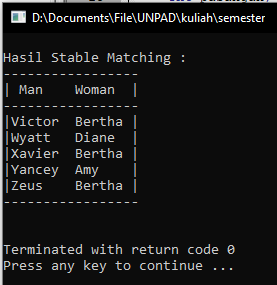
cout << "|" << pria[i].nama << "\t " << wanita[pria[i].pasangan].nama << "\t|" << endl;

}

cout << "-----------------" << endl;

}

* 1. **Hasil**



1. **Bagian Analisis di Modul Praktikum**

Analisis Algoritma Jawablah pertanyaan berikut:

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

|  |
| --- |
| Sama |

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 n2 iterasi menggunakan While Loop. Buktikan!

|  |
| --- |
| Karena, misal n = 10, maka jumlah operasi yang akan dieksekusi adalah sekitar 100. jadi untuk problem worksheet 01 ini, pada worst case nya akan terjadi operasi sebanyak 25 kali. |

Teorema (1.4): Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan. Buktikan!

|  |
| --- |
| Karena jumlah pria sama dengan jumlah wanita. Jadi ketika ada 4 pria yang telah memilki pasangan, pasti setidaknya ada satu wanita yang belum berpasangan. |

Teorema (1.5): Himpunan S yang dikembalikan saat terminasi adalah perfect matching Buktikan!

|  |
| --- |
| Betul, karena iterasi akan berakhir jika dan hanya jika ketika semua pria sudah berpasangan |

Teorema (1.6): Sebuah eksekusi algoritma G-S mengembalikan satu set pasangan S. Set S adalah pasangan yang stabil. Buktikan!

|  |
| --- |
| Dalam satu iterasi, pasti terjadi suatu matching yang stabil pada iterasi tersebut. Jika kondisi if tidak terpenuhi dan tidak menghasilkan stable matching, maka block else akan menghasilkan stable matching yang lain |