LAPORAN PRAKTIKUM I

ANALISIS ALGORITMA

Stable Matching Problem



Disusun oleh :

NAMA : Sharashena Chairani

Kelas : B

NPM : 140810180022

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

2020

# Studi Kasus

SMP ini dapat dilihat juga sebagai problem menyusun sistem dimana setiap pria dan wanita akhirnya bisa berpasangan.

Jadi pertimbangkan satu set M={n1, …, nn} dari n pria, dan satu set W={w1, …, wn} dari n wanita. Produk kartesius M x W menunjukkan set dari semua pasangan bentuk yang mungkin dipesan (n,w), di mana n Є M dan w Є W.

Matching S adalah seperangkat pasangan yang dipesan, masing-masing dari M x W, dengan properti yang masing-masing anggota M dan setiap anggota W muncul di paling banyak satu pasangan di S.

Dipandu oleh motivasi awal kita dalam hal pemberi kerja dan pelamar, kita harus khawatir tentang situasi berikut: Ada dua pasangan (n, w) dan (n′, w′) dalam S (seperti yang digambarkan pada Gambar 1.1) dengan properti bahwa n lebih suka w’ daripada w, dan w′ lebih suka n ke n′. Dalam hal ini, tidak ada yang bisa menghentikan n dan w′ meninggalkan pasangan mereka saat ini dan pergi bersama; set pernikahan menjadi tidak self-enforcing.

Tujuan kita adalah mengembalikan serangkaian pasangan tanpa ketidakstabilan (harus stabil).

Kita akan mengatakan bahwa S stabil jika

1. Perfect (1 laki-laki tepat berhubungan dengan satu perempuan), dan
2. tidak ada ketidakstabilan sehubungan dengan S

#### Syarat:

* Perfect Match: semua orang dicocokkan secara monogami.

1. Setiap pria mendapatkan satu wanita.
2. Setiap wanita mendapatkan satu pria.

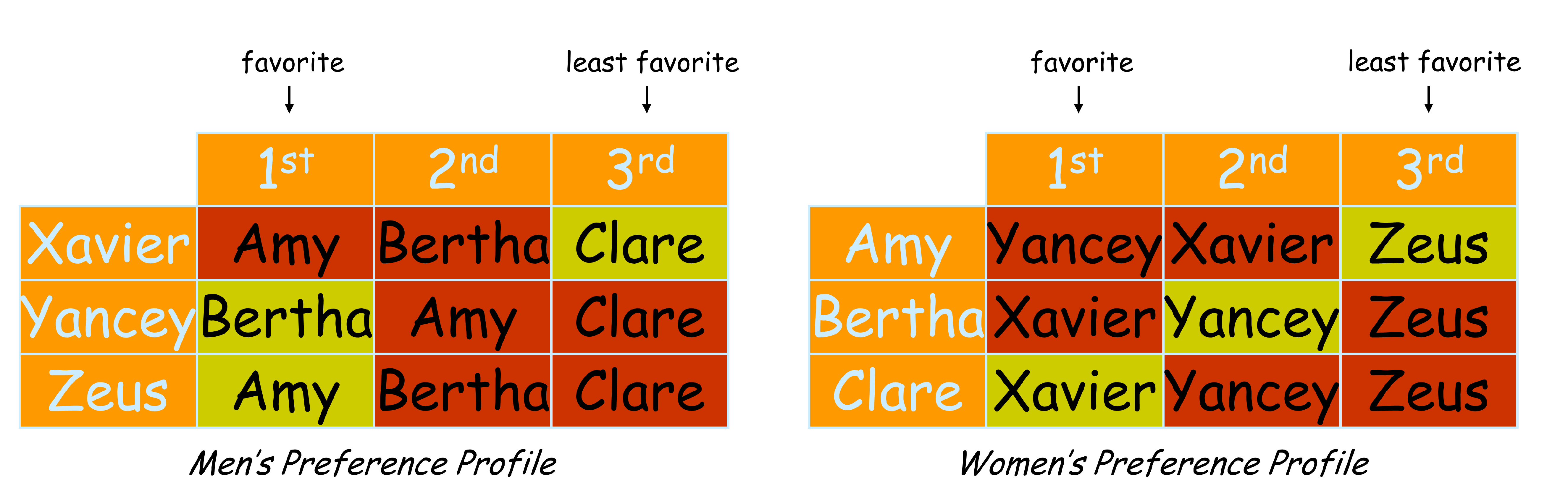
* Stable Matching: pencocokan sempurna tanpa pasangan tidak stabil.

#### Stable Matching Problemnya:

Dengan daftar preferensi pria dan wanita, temukan sebuah stable matching jika ada.

#### Contoh 1

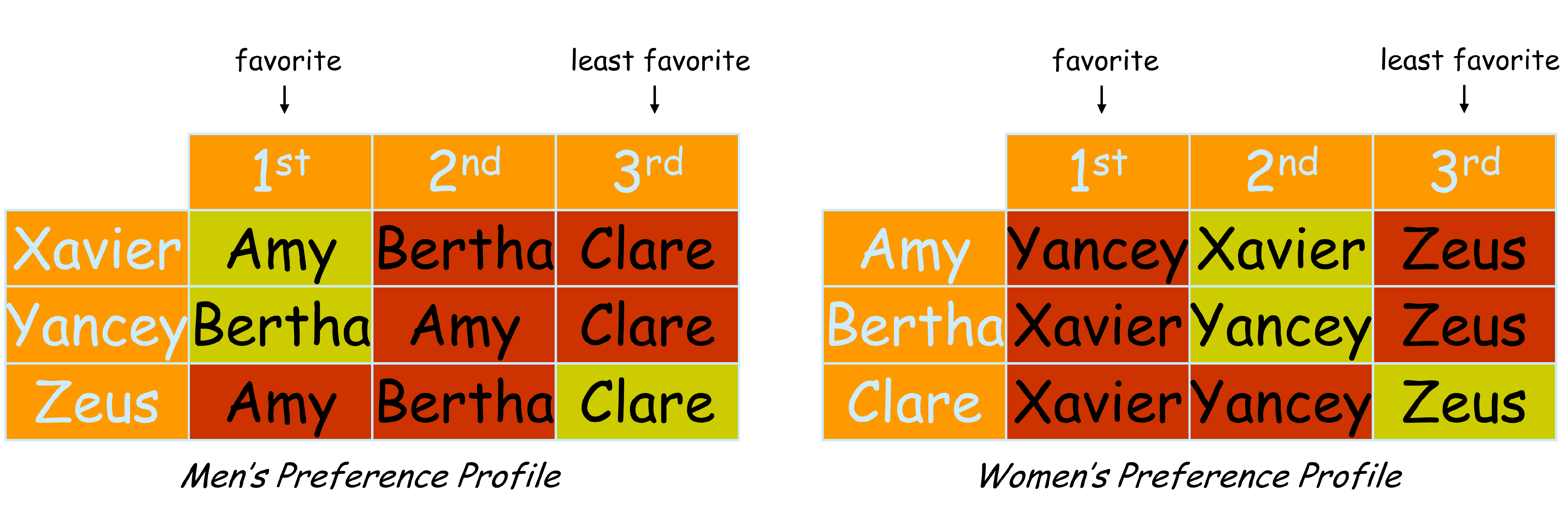
* + Pertanyaan: Jika dipasangkan X-C, Y-B, dan Z-A, apakah stabil?



* Jawaban: Tidak. Bertha & Xavier akan putus

#### Contoh 2

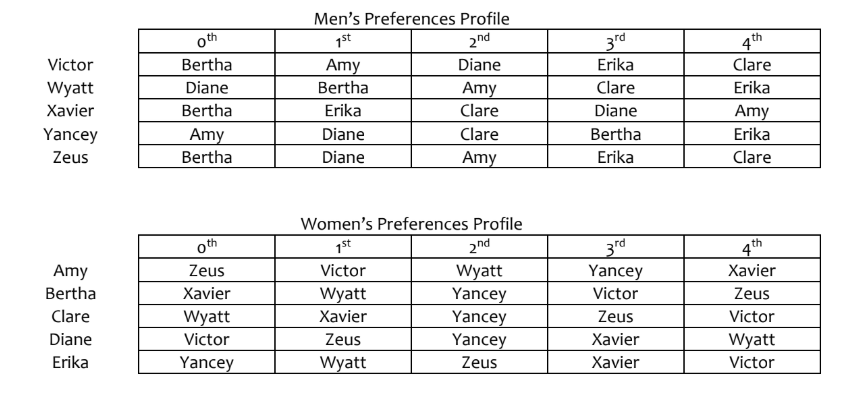
* + Pertanyaan: Jika dipasangkan X-A, Y-B, dan Z-C, apakah stabil?

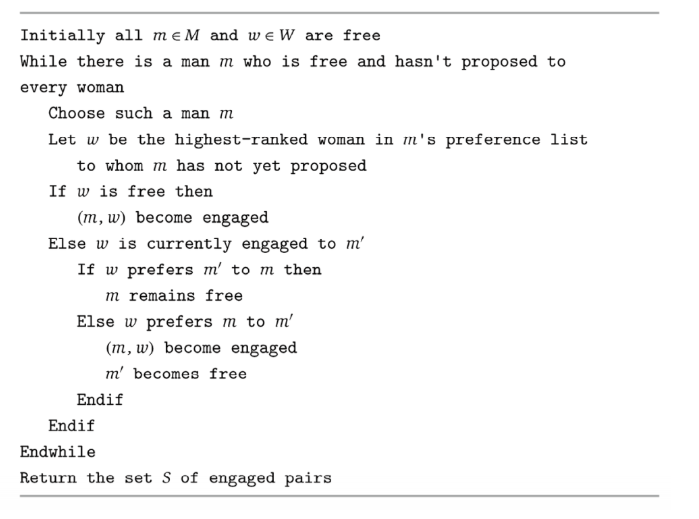


* + Jawaban: Ya

**Worksheet 01**

Dengan Algoritma Gale-Shapley, Kita dapat mencari Permasalahan himpunan *Stable-Matching* yang sesuai dengan *preference-lists* berikut ini. Gunakan processor terhebat yang Anda miliki (otak) untuk mengikuti algoritma G- S dan tidak perlu diuraikan per-looping tetapi Anda harus memahami hasil setiap looping.





**Tugas Praktikum**

* Ubahlah pseudocode algoritma G-S pada worksheet 01 ke dalam program menggunakan bahasa C++
* Gunakan table pria sebagai table acuan untuk memudahkan Anda menentukan pasangannya.
* Cocokkan jawaban Anda pada worksheet 01 dengan hasil program yang Anda buat
* Jika ada yang berbeda tuliskan bagian mana yang berbeda dan analisalah (Poin ini disampaikan pada bagian Analisis Algoritma) yang sudah disiapkan.

**Dengan Menggunakan C++**

**/\***

**Name : Sharashena Chairani**

**Class : B**

**NPM : 140810180022**

**Analisis Algoritma - Stable Matching Problem : dilihat dari Pria (Men Preference)**

**\*/**

**#include <iostream>**

**#include <string.h>**

**#include <stdio.h>**

**using namespace std;**

**#define N 5**

**bool PilihPria(int prefer[2\*N][N], int w, int m, int m1)**

**{**

**for (int i = 0; i < N; i++)**

**{**

**if (prefer[w][i] == m1)**

**return true;**

**if (prefer[w][i] == m)**

**return false;**

**}**

**}**

**void stableMatchingProb(int prefer[2\*N][N], string name[10])**

**{**

**int wanita[N];**

**bool priaFree[N];**

**memset(wanita, -1, sizeof(wanita));**

**memset(priaFree, false, sizeof(priaFree));**

**int jmlhJomblo = N;**

**while (jmlhJomblo > 0)**

**{**

**int m;**

**for (m = 0; m < N; m++)**

**if (priaFree[m] == false)**

**break;**

**for (int i = 0; i < N && priaFree[m] == false; i++)**

**{**

**int w = prefer[m][i];**

**if (wanita[w-N] == -1)**

**{**

**wanita[w-N] = m;**

**priaFree[m] = true;**

**jmlhJomblo--;**

**}**

**else**

**{**

**int m1 = wanita[w-N];**

**if (PilihPria(prefer, w, m, m1) == false)**

**{**

**wanita[w-N] = m;**

**priaFree[m] = true;**

**priaFree[m1] = false;**

**}**

**}**

**}**

**}**

**cout << "Sharashena C - 140810180022" << endl;**

**cout << endl;**

**cout << "~~~~~~~~~~~~~~~" << endl;**

**cout << " Pria Wanita" << endl;**

**cout << "~~~~~~~~~~~~~~~" << endl;**

**for (int i = 0; i < N; i++)**

**cout << " " << name[i+N] << "\t" << name[wanita[i]] << endl;**

**cout << "~~~~~~~~~~~~~~~" << endl;**

**}**

**int main()**

**{**

**string name[10] = {"Amy","Bertha","Clare","Diane","Erika", "Victor","Wyatt","Xavier","Yancey","Zeus"};**

**int prefer[2\*N][N] = {**

**{6, 5, 8, 9, 7},**

**{8, 6, 5, 7, 9},**

**{6, 9, 7, 8, 5},**

**{5, 8, 7, 6, 9},**

**{6, 8, 5, 9, 7},**

**{4, 0, 1, 3, 2},**

**{2, 1, 3, 0, 4},**

**{1, 2, 3, 4, 0},**

**{0, 4, 3, 2, 1},**

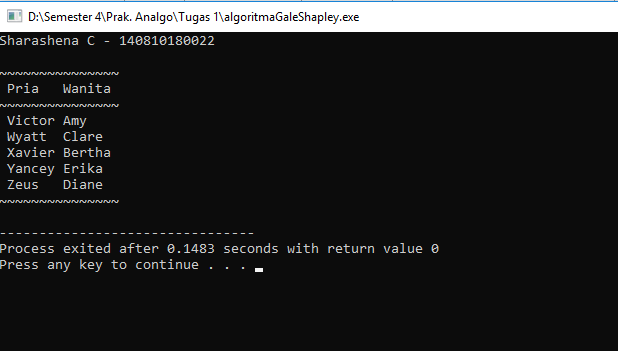
**{3, 1, 4, 2, 0},**

**};**

**stableMatchingProb(prefer, name);**

**return 0;**

**}**



**Iterasi 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

**Iterasi 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

**Iterasi 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

**Iterasi 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

**Iterasi 5**

**Victor - Amy**

**Wyatt - Clare**

**Xavier - Bertha**

**Yancey - Erika**

**Zeus - Diane**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

Jawablah pertanyaan berikut :

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

Ya, sama

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

#### Fakta (1.1) :

Setiap wanita tetap bertunangan dari titik di mana dia menerima Lamaran pertamanya; dan urutan Pasangan yang bertunangan dengannya menjadi lebih baik dan lebih baik lagi (hal ini sesuai dengan daftar Woman Preferences).  tidak perlu dipertanyakan

#### Fakta (1.2) :

Urutan wanita yang dilamar pria lebih buruk dan lebih buruk lagi (hal ini sesuai dengan daftar Men Preferences).  tidak perlu dipertanyakan

#### Teorema (1.3):

Algoritma G-S berakhir setelah paling banyak n2 iterasi menggunakan While Loop. Buktikan!

Pada saat While Loop Setiap Pasangan dicoba untuk dicocokan, yaitu dimana Pria yang FREE atau Lajang Melamar Wanita berikutnya dalam daftar pilihannya, lalu Ke seseorang yang belum pernah dilamar oleh Pria tersebut sebelumnya.

Karena Pria berjumlah n dan setiap daftar prefensi panjangnya adalah n, maka paling banyak proposal adalah n2. Jadi jumlah dalam iterasi paling banyak adalah n2

#### Teorema (1.4):

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

Buktikan!

Misalkan ada waktu tertentu dalam pelaksanaan algoritma ketika seorang Pria FREE/Lajang, namun telah melamar Setiap Wanita yang ada. Maka hal tersebut berarti, setiap Wanita telah dilamar setidaknya satu kali, dan setiap wanita bertunangan.

Jadi, kita telah melibatkan n Wanita dan karenanya n Pria bertunangan, yang menyiratkan bahwa m juga terlibat dan tidak mungkin ada Pria yang tidak memiliki Pasangan karena jumlah pasangannya sama-sama berjumlah n.

#### Teorema (1.5):

Himpunan S yang dikembalikan saat terminasi adalah *perfect matching* Buktikan!

Iya Benar, karena setiap Pria berpasangan dengan setiap Wanita

#### Teorema (1.6):

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

Buktikan!

Eksekusi Algoritma Gale Shalpey menunjukkan bahwa pencocokan yang dikembalikan adalah pencocokan sempurna. Buktinya dengan kontradiksi.

Misalkan tidak, maka ada seorang Pria yang masih FREE di akhir algoritma. Menurut teori 2, itu berarti m belum melamar beberapa Wanita. Tetapi kemudian, algoritma tidak akan keluar dari pengulangan loop, menghasilkan kontradiksi yang diinginkan.

Hal itu, Menunjukkan bahwa pencocokan yang dikembalikan stabil. Maka, buktinya dengan kontradiksi.

Misalkan ada Pria m dan m’ dan Wanita w dan w’ sehingga (m, w) dan (m’, w’) berada di S, tetapi m lebih suka w’ ke w dan w’ lebih suka m ke m’.

Dengan algoritma, w adalah Wanita terakhir yang diajukan. Karena m lebih suka w’ daripada w, m harus sudah mengusulkan ke w’ sebelum usulannya ke w. Pada saat itu, atau nanti, w’ bertunangan dengan seorang Pria, Misal m’’, yang ia sukai lebih dari m.

Pada akhirnya, w’ bertunangan dengan m’. Oleh teori 1, ditemukan bahwa w’ lebih memilih m’ daripada m’’ dan lebih memilih m’’ daripada m;

Hal tersebut menunjukkan bahwa w’ lebih suka m’ daripada m, bertentangan dengan asumsi bahwa w’ lebih memilih m daripada m’.