**Laporan Praktikum Jaringan Komputer**

**Modul Praktikum 1**



**Disusun Oleh:**

Fadhli Hibatul Haqqi (140810170019)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2018/2019**

**Worksheet 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| Victor | Bertha | Amy | Diane | Erica | 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 |

Men’s Preferences :

Women’s Preferences :

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

Hasil Stable Matching :

Victor, Amy;

Wyatt, Clare;

Xavier, Bertha;

Yancey, Erika;

Zeus, Diane;

**Program**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nama : Fadhli Hibatul Haqqi

NPM : 140810170019

Kelas : A

Tanggal : 01/03/2019s

Program : G-S Algorithm for Stable Match

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<string>

using namespace std;

typedef int prefer[5];

struct manusia{

string nama;

int pasangan;

int preferences[5];

};

manusia inputOrang(string nama,prefer prefers){

manusia mns;

mns.nama = nama;

mns.pasangan = -1;

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

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

}

return mns;

}

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

bool adaBebas = true;

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

while(adaBebas == true){

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;

}

}

adaBebas = false;

j=0;

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

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

i = l;

adaBebas = true;

break;

}

}

}

}

int main(){

manusia pria[5];

manusia wanita[5];

prefer prefers;

//Pria

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

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

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

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

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

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

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

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

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

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

//Wanita

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

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

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

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

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

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

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

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

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

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

stableMatching(pria,wanita);

//output

cout<<pria[0].nama<<", "<<wanita[pria[0].pasangan].nama<<endl;

cout<<pria[1].nama<<", "<<wanita[pria[1].pasangan].nama<<endl;

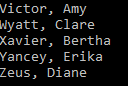
cout<<pria[2].nama<<", "<<wanita[pria[2].pasangan].nama<<endl;

cout<<pria[3].nama<<", "<<wanita[pria[3].pasangan].nama<<endl;

cout<<pria[4].nama<<", "<<wanita[pria[4].pasangan].nama;

}

**Screenshot :**



**Analisis Algoritma**

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

Jawab :

Ya. Sama persis.

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

Jawab :

Ya contohnya pada worksheet 1 kasus terburuk adalah ketika setiap iterasi hanya satu pria yang tidak putus putus dengan pasangannya. Sehingga harus mengulang mencocokan sebanyak n kali. Maka iterasi paling banyak adalah n2 kali.

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

Jawab :

Karena setiap pria atau wanita harus monogami, maka jumlah wanita bebas sama dengan jumlah pria bebas.

1. Himpunan S yang dikembalikan saat terminasi adalah perfect matching. Buktikan!

Jawab :

Ya, contoh pada jawaban worksheet 1 menghasilkan pasangan

Victor, Amy;

Wyatt, Clare;

Xavier, Bertha;

Yancey, Erika;

Zeus, Diane;

Setiap pria dan wanita monogami, dan tidak ada pria atau wanita yang tidak memiliki pasangan. Sehingga terbukti himpunan S yang dikembalikan adalah perfect matching.

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

Jawab :

Syarat stable match:

1. Harus perfect match
2. Setiap pasangan set S tidak ada ketidakstabilan

Dari jawaban no 4 dapat diketahui bahwa syarat (1) terpenuhi. Algoritma G-S memasangkan pasangan sesuai dengan keinginannya, sehingga setiap pasangan pada set S tidak ada ketidakstabilan, maka syarat (2) terpenuhi. Dengan terpenuhinya kedua syarat stable match, maka terbukti bahwa set S yang dihasilkan dari algoritma G-S merupakan stable match.