Tugas Praktikum Analisis Algoritma

ALGORITMA GALE-SHAPLEY



Dibuat Oleh :

Adryan Luthfi Faiz 140810160049

**UNIVERSITAS PADJADJARAN**

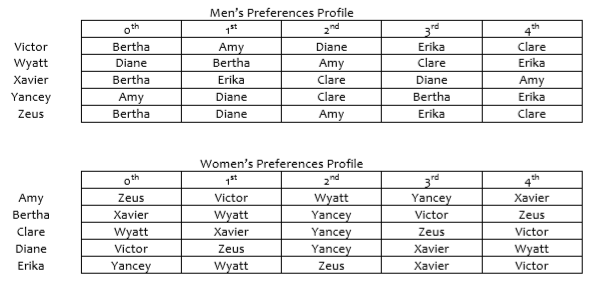
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

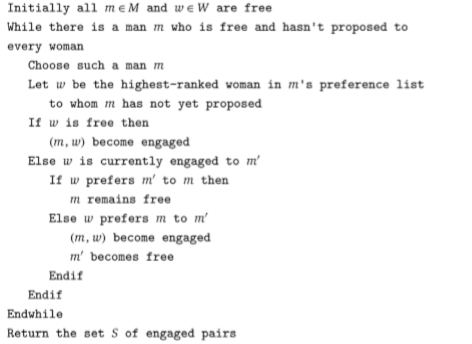
**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**2018**

WORKSHEET 01

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 output 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.

PROGRAM C++

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

Nama : Adryan Luthfi Faiz

NPM : 140810160049

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

#include <iostream>

using namespace std;

bool statusFree (bool\*Status){ // mengecek status setidaknya ada 1 orang yang Free

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

{

if(\*(Status+i)==true) return true;

}

return false;

}

int\* Matching(int\*\*MP,int\*\*WP) { // Matching List

bool isManFree[5],isWomanFree[5],isManProposed[5][5]; // to represent matching states

int \*match = new int[5]; // // nilai index (man-woman)

for(int i=0;i<5;i++) // inisialisasi

{

isManFree[i]=true;

isWomanFree[i]=true;

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

isManProposed[i][j]=false;

}

match[i]=-1;

}

while(statusFree(isManFree)) // mencocokkan ketika returns false.

{

int indexM;

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

{

if(isManFree[i]==true) {

indexM=i; break;

} // mencocokkan

}

int indexWo;

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

{

int w=MP[indexM][i];

if(isManProposed[indexM][w]==false) {

indexWo=w; break;

} // prioritas setelahnya

}

isManProposed[indexM][indexWo]=true;

if(isWomanFree[indexWo])

{

isManFree[indexM]=false;

isWomanFree[indexWo]=false;

match[indexM]=indexWo; // mereka cocok

}

else

{

int indexRival; // menyimpan yang memiliki index yang sama

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

{

if(match[i]==indexWo){

indexRival=i;

break;

}

}

int pM,pRival;

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

{

if(WP[indexWo][i]==indexM) pM=i;

if(WP[indexWo][i]==indexRival) pRival=i;

}

if(pM<pRival) // perempuan memilih

{

isManFree[indexM]=false;

isManFree[indexRival]=true;

isWomanFree[indexWo]=false;

match[indexM]=indexWo; // mengganti ke yang cocok

}

}

}

return match;

}

void listName(){

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

cout<<"1 = Victor" <<endl <<"2 = Wyatt" <<endl <<"3 = Xavier" <<endl <<"4 = Yancey" <<endl <<"5 = Zeus" <<endl<<endl;

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

cout<<"1 = Amy" <<endl <<"2 = Bertha" <<endl <<"3 = Clare" <<endl <<"4 = Diane" <<endl <<"5 = Erika" <<endl;

}

void NameMP(int n){

switch(n){

case 1 :cout <<"Victor"; break;

case 2 :cout <<"Wyatt"; break;

case 3 :cout <<"Xavier"; break;

case 4 :cout <<"Yancey"; break;

case 5 :cout <<"Zeus"; break;

}

}

void NameWP(int n){

switch(n){

case 1 :cout <<"Amy"; break;

case 2 :cout <<"Bertha"; break;

case 3 :cout <<"Clare"; break;

case 4 :cout <<"Diane"; break;

case 5 :cout <<"Erika"; break;

}

}

main (){

int\*\*MP,\*\*WP;

MP=new int\*[5];

listName();

cout<<endl<<"Masukkan Data Pria (nomor 1 - 5) : "<<endl;

for(int i=0;i<5;i++) // input data

{

MP[i]=new int[5];

NameMP(i+1);

cout<<"\t";

for(int j=0;j<5;j++)

{

cin>>MP[i][j];

}

}

cout<<endl;

WP=new int\*[5];

cout<<endl<<"Masukkan Data Wanita (nomor 1 - 5) : "<<endl;

for(int i=0;i<5;i++) // input data

{

WP[i]=new int[5];

NameWP(i+1);

cout<<"\t";

for(int j=0;j<5;j++)

{

cin>>WP[i][j];

}

}

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

{

for(int j=0;j<5;j++)

{

MP[i][j]--; // inputs are 1~n, get indexes 0~n-1

WP[i][j]--;

}

}

int\*match=Matching(MP,WP);

cout<<endl;

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

{

cout<<i + 1 << " <> " <<\*(match+i)+1<<endl; // output: matching result

}

cout<<endl<<"Convert To Name"<<endl<<endl;

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

{

NameMP(i+1);

cout<< " <> " ;

NameWP(\*(match+i)+1);

cout<<endl; // output: matching result

}

delete [] match;

return 0;

}

/\* Data Tugas Praktikum 1

Men's Preferences Profile

Victor 2 1 4 5 3

Wyatt 4 2 1 3 5

Xavior 2 5 3 4 1

Yancey 1 4 3 2 5

Zeus 2 4 1 5 3

Women's Preferences Profile

Amy 5 1 2 4 3

Bertha 3 2 4 1 5

Clare 2 3 4 5 1

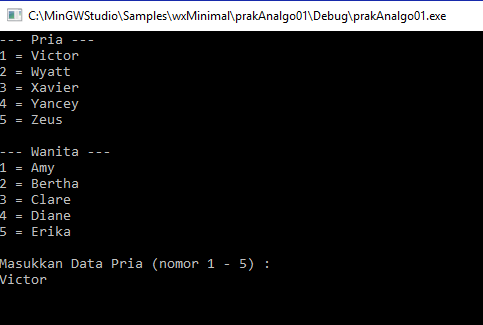
Diane 1 5 4 3 2

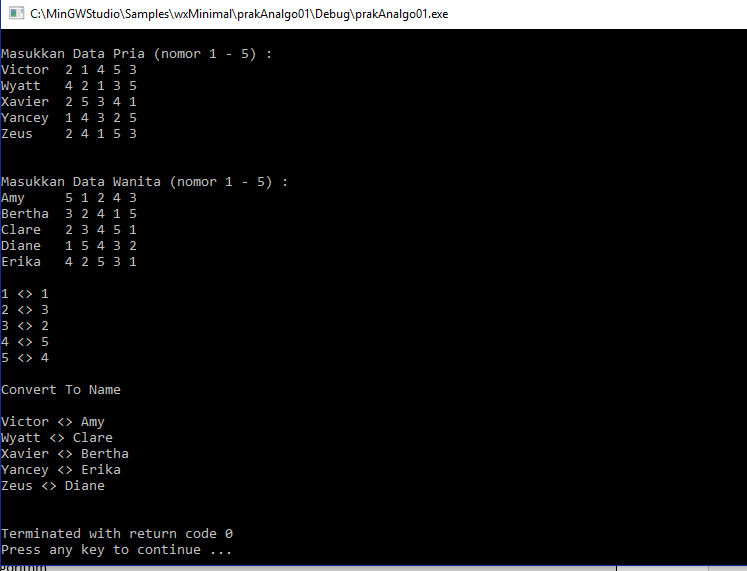
Erika 4 2 5 3 1

\*/

// Referensi : https://stackoverflow.com/questions/12858734/c-implementation-of-gale-shapley-algorithm

Screenshot





Analisis

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

🡪 Iya Sama Persis

* 1. 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!

🡪 Setiap pria akan memilih satu wanita yang belum pernah ia pilih sebelumnya. Karena jumlah pria sebanyak n dan setiap pria memiliki preferensi sebanyak n jumlah wanita, maka jumlah iterasi yang dapat terjadi 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!

🡪 Dapat dibuktikan dengan kontradiksi. Setiap pria memiliki seluruh list wanita dan begitu juga wanita. Berarti tidak mungkin ada orang yang masih single jika setiap n sudah berpasangan.

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

🡪 Pria pasti hanya akan melamar apabila belum berpasangan atau pasangan sebelumnya tidak cocok. Sedangkan wanita akan selalu memilih pria dengan preferensi teratas untuk bertunangan dengannya. Baik pria dan wanita, prinsip mereka adalah melamar sesuai urut preferensi. Dengan itu Himpunan S adalah perfect matching dikarenakan teori tersebut.

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

🡪 Tidak ada pria yang bisa ditolak oleh semua wanita. Wanita pun harus mendapatkan satu pria dan ia hanya dapat menolak lamaran ketika ada pria yang lebih tinggi preferensinya dibandingkan pria sebelumnya. Setiap iterasi dari loop sementara melibatkan tepat satu proposal dan pria tidak akan melamar wanita yang sama dua kali. Dalam kasus terburuk, seorang pria bisa melamar semua wanita hingga ia mendapatkan pasangannya. Dengan menggunakan algoritma G-S, dapat kita simpulkan bahwa algoritma akan berakhir setelah semua mendapat pasangan. Selain itu pasangan yang sudah dipasangkan juga harus menunjukkan kecocokan dengan preferensi masing-masing dan pencocokan algoritma G-S dianggap selalu stabil.