**Tugas Praktikum**

**Analisis Algoritma**

**(*Stable Matching Problem - Gale Shapley Algorithm*)**



Disusun oleh :

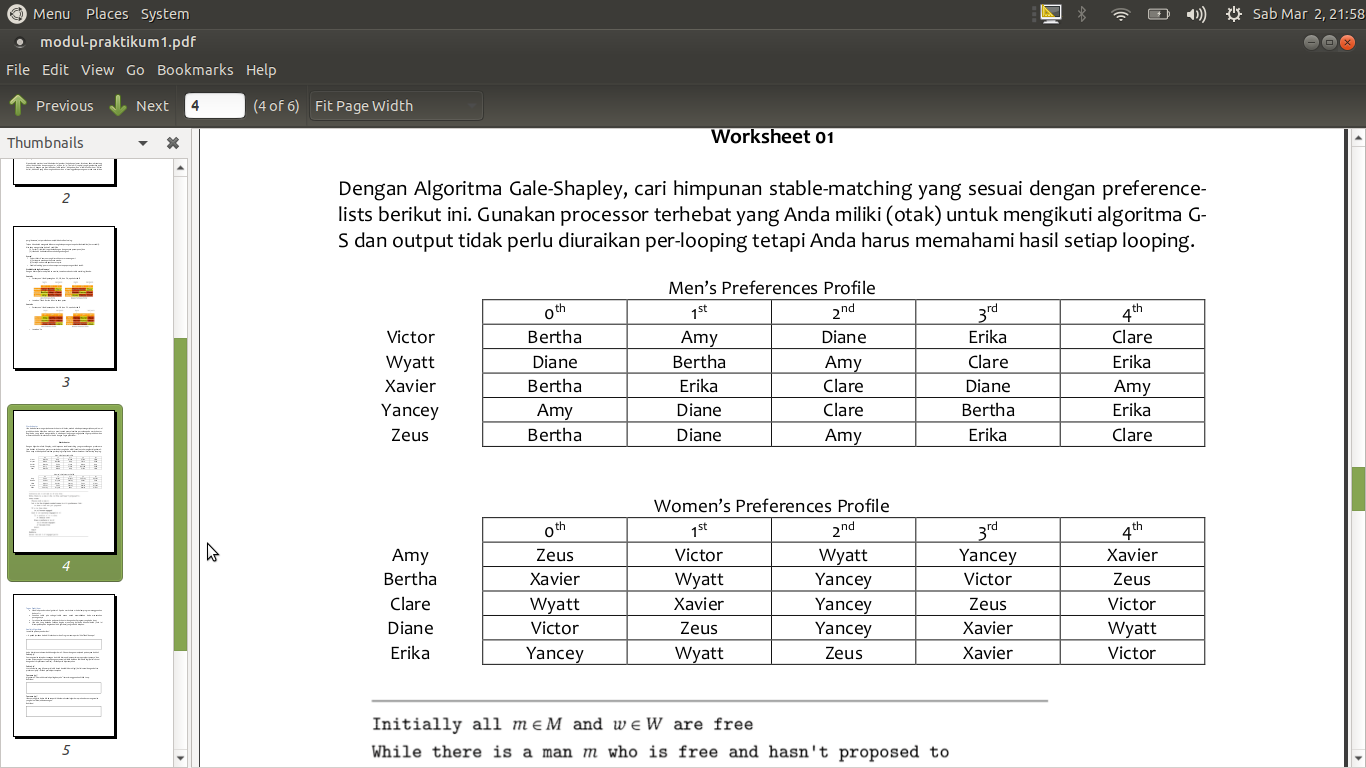
Baby Cattleya Gustina Permatagama

(140810160048)

S-1 Teknik Informatika   
Fakultas Matematika & Ilmu Pengetahuan Alam   
Universitas Padjadjaran  
Jalan Raya Bandung - Sumedang Km. 21 Jatinangor 45363

***Stable Matching Problem***

Dengan Algoritma Gale-Shapley, cari himpunan *stable-matching* yang sesuai dengan *preference-lists* berikut ini.



**Jawab :**

Proses *Matchmaking* :

* Victor dengan Bertha
* Wyatt dengan Diane
* **Xavier dengan Bertha**, Bertha lebih memilih Xavier dan memutuskan Victor.
* Yancey dengan Amy
* Zeus ingin dengan Bertha, namun Bertha lebih memilih Xavier. Zeus sendiri
* **Victor dengan Amy**, Amy memutuskan Yancey.
* Yancey dengan Diane. Diane putus dengan Wyatt
* **Zeus dengan Diane**. Diane putus dengan Yancey
* **Wyatt dengan Clare**
* **Yancey dengan Erika**

**Jadi didapat *stable matching* : Xavier dan Bertha, Victor dan Amy, Zeus dengan Diane, Wyatt dengan Clare, Yancey dengan Erika.**

Program C++ :

/\*

Nama Program : Algoritma Gale Shapley

Oleh : Baby Cattleya G.P.

Dibuat : Minggu, 3 Maret 2019

\*/

#include <algorithm>

#include <iostream>

#include <map>

#include <queue>

#include <string>

#include <vector>

using namespace std;

const char \*men\_data[][6] = {

{ "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" }

};

const char \*women\_data[][6] = {

{ "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" }

};

typedef vector<string> PrefList;

typedef map<string, PrefList> PrefMap;

typedef map<string, string> Couples;

// Does 'first' appear before 'second' in preference list?

bool prefers(const PrefList &prefer, const string &first, const string &second)

{

for (PrefList::const\_iterator it = prefer.begin(); it != prefer.end(); ++it)

{

if (\*it == first) return true;

if (\*it == second) return false;

}

return false; // no preference

}

int main()

{

PrefMap men\_pref, women\_pref;

queue<string> bachelors;

// init data structures

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

{

for (int j = 1; j < 6; ++j) // preference

{

men\_pref[ men\_data[i][0]].push\_back( men\_data[i][j]);

women\_pref[women\_data[i][0]].push\_back(women\_data[i][j]);

}

bachelors.push(men\_data[i][0]);

}

Couples engaged; // <woman,man>

cout << "Proses Matchmaking:\n";

while (!bachelors.empty())

{

const string &suitor = bachelors.front();

const PrefList &preflist = men\_pref[suitor];

for (PrefList::const\_iterator it = preflist.begin(); it != preflist.end(); ++it)

{

const string &bride = \*it;

if (engaged.find(bride) == engaged.end()) // she's available

{

cout << "\t" << bride << " dan " << suitor << "\n";

engaged[bride] = suitor; // hook up

break;

}

const string &groom = engaged[bride];

if (prefers(women\_pref[bride], suitor, groom))

{

cout << "\t" << bride << " memutus " << groom << " untuk " << suitor << "\n";

bachelors.push(groom); // dump that zero

engaged[bride] = suitor; // get a hero

break;

}

}

bachelors.pop(); // pop at the end to not invalidate suitor reference

}

cout << "Hasil Pasangan\n";

for (Couples::const\_iterator it = engaged.begin(); it != engaged.end(); ++it)

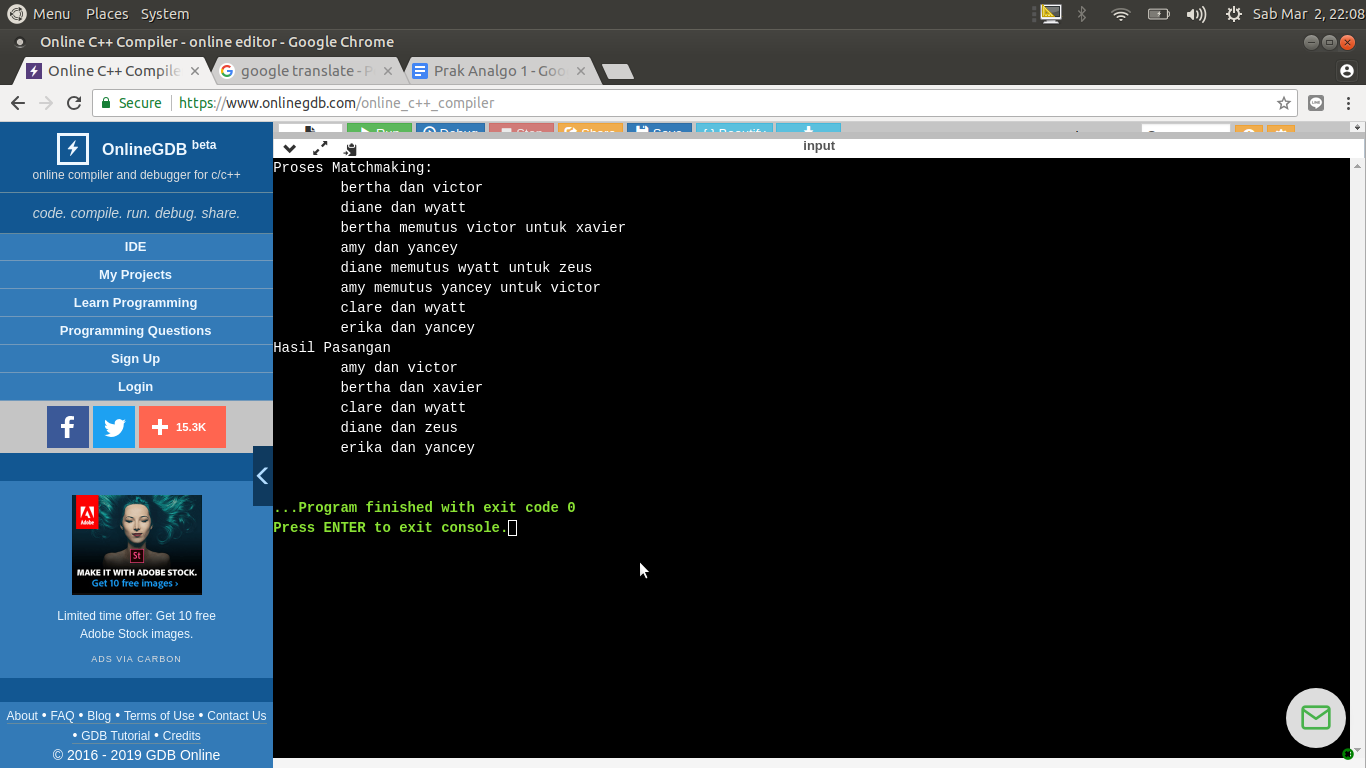
{

cout << "\t" << it->first << " dan " << it->second << "\n";

}

}

Hasil Program :



Jawaban *worksheet* dan program cocok, tidak ada perbedaan.

Analisis Algoritma

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

**Jawab :**

*Ya, jawaban Saya di Worksheet 01 dan Program sama persis.*

1. Teorema (1.3):

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

**Jawab :**

*Di setiap iterasi while loop, seorang pria lajang melamar wanita berikutnya dalam daftar pilihannya, seseorang yang tidak pernah dia lamar sebelumnya. Karena ada n laki-laki dan setiap daftar preferensi memiliki panjang n, ada paling banyak n2 lamaran yang bisa terjadi. Jadi jumlah iterasi yang dapat terjadi paling banyak adalah n2*

1. Teorema (1.4):

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

**Jawab :**

Membuktikan dengan kontradiksi :

*Misalkan ada waktu tertentu dalam eksekusi algoritma ketika seorang pria m bebas, namun telah melamar setiap wanita. Ini berarti bahwa, setiap wanita telah dilamar setidaknya satu kali. Padahal setiap wanita akan memiliki pasangan jika telah dilamar satu kali. Kita punya n wanita terlamar yang berarti n pria yang telah memiliki pasangan, jadi kontradiksi Pernyataan di teorema benar.*

1. Teorema (1.5):

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

**Jawab :**

Membuktikan dengan kontradiksi :

*Misalkan ada pria m yang belum memiliki pasangan saat terminasi. Karena ada n pria dan n wanita, maka ada wanita w yang belum memiliki pasangan juga. Ketika seorang wanita memiliki pasangan, ia tidak pernah sendiri lagi, ia hanya bertukar pasangan. Berarti jika w sendiri maka belum ada yang melamar w. Padahal m seharusnya telah melamar semua wanita (maka kontradiksi). Jadi asumsi diatas benar.*

1. Teorema (1.6):

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

Jawab :

Asumsi lewat kontrakdiksi bahwa algoritma G-S memproduksi sebuah *unstable matching.* Misal telah ada pasangan (m,w’) dengan preferensi

* m prefers w' daripada w
* w' prefers m daripada m'.

Maka m melamar w' sebelum melamar w.

Tapi seorang wanita hanya bisa menolak pria jika ia mendapat lamaran lagi dari pria yang lebih dia inginkan. Jika w' prefers m' daripada m, adalah kontradiksi. Jadi G-S algorithm benar memproduksi *stable matching.*