**LAPORAN**

**PRAKTIKUM ANALISIS ALGORITMA**

**TUGAS 01**



**Disusun Oleh:**

Dimas Satria Prakoso (140810170007)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**SUMEDANG**

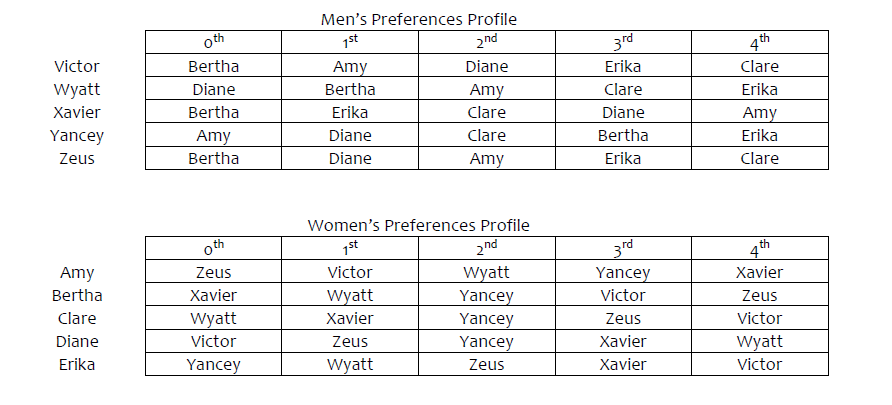
**2019**

**LAPORAN**

**PRAKTIKUM ANALISIS ALGORITMA**

**TUGAS 01**

**Worksheet 01**



Dengan algoritma Gale-Shapley, didapat himpunan Stable Matching sebagai berikut :

* Victor dan Amy
* Wyatt dan Clare
* Xavier dan Bertha
* Yancey dan Erika
* Zeus dan Diane

**Bentuk programnya**

Dengan menggunakan bahasa C++ , maka didapat program Stable Matching: Marriage Problem ini sebagai berikut

/\*

Nama Program    : Stable Matching: Marriage Problem

Nama            : Dimas Satria Prakoso

NPM             : 140810170007

Tanggal Buat    : 5 Maret 2019

Praktikum : Analisis Algoritma

Tugas : Tugas 1 Analgo

Deskripsi       :

.cpp tested with CodeBlocks 16.01

\*/

#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"},

{ "Yance", "Amy","Diane","Clare","Bertha","Erika"},

{ "Zeus", "Bertha","Diane","Amy","Erika","Clare"}

};

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

{ "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"}

};

typedef vector<string> PrefList;

typedef map<string, PrefList> PrefMap;

typedef map<string, string> Couples;

// Apakah 'first' datang lebih dulu sebelum 'second' di preference list-nya?

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;

cout<<"Stable Matching: Marriage Problem\ndisusun oleh: Dimas Satria\n\n";

// masukan data

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 << "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()) // Jika cewenya available

{

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

engaged[bride] = suitor; // Jadian

break;

}

const string &groom = engaged[bride];

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

{

cout << "\t" << bride << " mencampakan " << groom << " karena " << suitor << "\n";

bachelors.push(groom); // Masukan kembali groom menjadi bujangan

engaged[bride] = suitor; // Bride berpasangan dengan suitor

break;

}

}

bachelors.pop();

}

cout << "Pernikahan:\n";

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

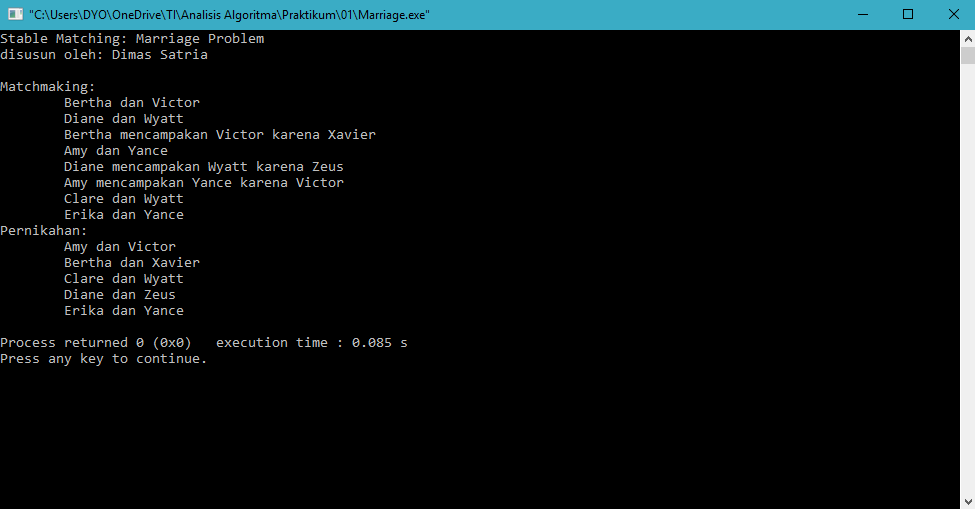
{

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

}

}

Dengan hasil sebagai berikut



**Analisis Algoritma**

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

Jawabannya sama.

1. **Teorema (1.3):** Algoritma G-S berakhir setelah paling banyak n2 iterasi menggunakan While Loop. Buktikan!

Jika banyak laki-laki = banyak perempuan = n dan kasus terburuknya ialah ketika semua laki-laki melamar pada semua perempuan maka worst case algoritma ini ialah : n x n = 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!

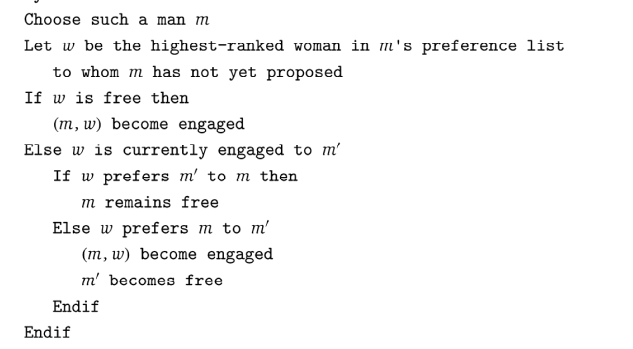
Kerena jumlah laki-laki dan wanita sama dan satu laki-laki hanya dapat bertunangan dengan satu perempuan maka jelas ketika ada seorang pria bebas maka minimal ada satu orang wanita yang belum berpasangan pula.

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

hasil yang akan dihasilkan dari algoritma diatas ialah pasangan-pasangan. setiap pria akan selalu mendapatkan sorang wanita dan wanita akan selalu mendapatkan seorang pria. Didukung dengan teoram 1.4 dan kenyataan dari algoritma ini yang akan selalu menghasilkan pasangan yang monogami maka dapat disimpulkan Himpunan S adalah perfect matching

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

Himpunan S dikatan non-stable matching jika dan hanya jika Himpunan S memiliki pasangan yang tidak stabil. Hal ini kontradiksi dengan algoritma awal kita dimana (lanjut pada halaman selanjutnya)



Dari algoritma diatas disimpulkan bahwa seorang pria pasti akan melamar wanita sesuai dengan list preferensi nya dan seorang wanita hanya bisa menolak atau menerima juga sesuai dengan list preferensinya. Hal ini menyatakan bahwa list preferensinya akan selalu sinkron, maka pasangan yang didapat merupakan pasangan terbaik yang stabil. Teorema(1.6) : Himpunan S merupakan pasangan yang stabil ini terbukti.