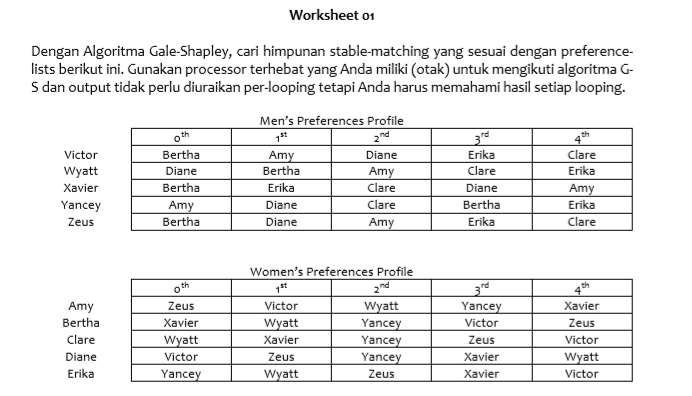
Laporan Praktikum 1  
Praktikum Analisis Algoritma



Imron Madani (140810170061)

Fakultas Matematika dan Ilmu Pengetahuan Alam  
Program Studi Teknik Informatika  
Universitas Padjajaran  
2019



Maka akan didapat pasangan yang sesuai yaitu

Victor - Amy

Wyatt - Clare

Xavier - Bertha

Yancey - Erika

Zeus – Diane

**Program Bahasa C++**

|  |
| --- |
| #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\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 << "Mencocokkan:\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 cewek jomblo  {  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";  }  } |

**Analisis Algoritma**

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

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

1. 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
2. Fakta (1.2): Urutan wanita yang dilamar pria lebih buruk dan lebih buruk lagi (hal ini sesuai dengan daftar preferensi pria). -> tidak perlu dipertanyakan
3. Teorema (1.3): Algoritma G-S berakhir setelah paling banyak n2 iterasi menggunakan While Loop. Buktikan!

Jika n=jumlah laki-laki=jumlah perempuan maka pada kasus terburuk setelah sebanyak , loop akan berhenti karena sudah melakukan semua percobaan pencocokan.

1. Teorema (1.4): Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan. Buktikan!

Jika jumlah pria=jumlah wanita, dan setiap orang hanya mempunyai satu pasangan, maka akan ada wanita yang belum dia ajak bertunangan.

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

Himpunan S adalah perfect matching, maka program tidak akan berhenti sebelum pencocokannya stabil.

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

Pada awal algoritma akan dilakukan pencocokan berdasarkan preferensi pria. kemudian jika ada lamaran lagi maka akan dicek berdasarkan preferensi wanita. Program akan terus diulang sampai para pria mendapatkan pasangan yang stabil. Maka set S merupakan pasangan yang stabil