**LAPORAN PRAKTIKUM**

**MODUL 1 – STABLE MATCHING PROBLEM**

**PRAKTIKUM MATA KULIAH ANALISIS ALGORITMA**



Disusun oleh:

Nama : Muhammad Fadillah Arsa

NPM : 140810170005

Kelas : A

Program Studi : S1 Teknik Informatika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2019**

**LAPORAN PRAKTIKUM**

**MODUL 1 – STABLE MATCHING PROBLEM**

**PRAKTIKUM MATA KULIAH ANALISIS ALGORITMA**

# PENDAHULUAN

Stable Matching Problem (SMP) adalah problem algoritmik yang memberikan ilustrasi mengenai berbagai tema yang dipelajari di analisis algoritma ini. Algoritma ini muncul dari beberapa problem praktis. Oleh karena itu supaya problemnya jelas dan penyelesaian tepat perlu dilakukan 3 langkah berikut:

* Mencermati problem
* Memformulasikan problem
* Mendesain algoritma

Stable Matching Problem berasal, sebagian, pada tahun 1962, ketika David Gale dan Lloyd Shapley, dua matematika ekonom, mengajukan pertanyaan:

***Bisakah seseorang merancang sebuah perguruan tinggi proses penerimaan, atau proses perekrutan pekerjaan, itu mandiri (otomatis)?***

Inti dari proses aplikasi adalah interaksi antara dua jenis pihak yang berbeda: **perusahaan dan pelamar**.

Setiap pelamar memiliki daftar preferensi perusahaan yang ingin dimasuki, dan setiap perusahaansetelah aplikasi masuk-membentuk daftar preferensi akan pelamarnya. Berdasarkan preferensi ini, perusahaan memberikan penawaran kepada beberapa pelamar mereka, pelamar memilih penawaran mana yang akan mereka terima.

Bagaimana jika tidak dilakukan secara otomatis? Kemungkinan resiko kecurangan tinggi.

Jadi inilah pertanyaan yang diajukan Gale dan Shapley: Diberikan seperangkat preferensi di antara pemberi kerja dan pelamar, dapatkah kami menetapkan pelamar untuk pemberi kerja sehingga untuk setiap pemberi kerja E, dan setiap pelamar A yang tidak dijadwalkan bekerja untuk E, setidaknya satu dari dua hal berikut ini yang terjadi?

1. E lebih memilih setiap satu dari daftar pelamar yang diterima(A); atau
2. A lebih suka situasinya saat ini daripada bekerja untuk pemberi kerja E

Jika ini berlaku, hasilnya stabil: kepentingan pribadi individu akan mencegah kesepakatan pemohon/pemberi kerja dibuat dibalik layar. Gale dan Shapley mengembangkan solusi algoritmik yang tajam untuk problem ini, yang akan kita pelajari.

# STUDI KASUS

SMP ini dapat dilihat juga sebagai problem menyusun sistem dimana setiap pria dan wanita akhirnya bisa berpasangan.

Jadi pertimbangkan satu set ={ 1, …, } dari pria, dan satu set W={ 1, …, } dari wanita. Produk kartesius menunjukkan set dari semua pasangan bentuk yang mungkin dipesan ( , ), di mana Є dan Є .

Matching adalah seperangkat pasangan yang dipesan, masing-masing dari , dengan properti yang masing-masing anggota dan setiap anggota muncul di paling banyak satu pasangan di .

Dipandu oleh motivasi awal kita dalam hal pemberi kerja dan pelamar, kita harus khawatir tentang situasi berikut: Ada dua pasangan ( , ) dan ( ′, ′) dalam (seperti yang digambarkan pada Gambar 1.1) dengan properti bahwa lebih suka ’ daripada , dan ′ lebih suka ke ′. Dalam hal ini, tidak ada yang bisa menghentikan dan ′ meninggalkan pasangan mereka saat ini dan pergi bersama; set pernikahan menjadi tidak self-enforcing.

Tujuan kita adalah mengembalikan serangkaian pasangan tanpa ketidakstabilan (harus stabil).

Kita akan mengatakan bahwa S stabil jika

1. Perfect (1 laki-laki tepat berhubungan dengan satu perempuan), dan
2. tidak ada ketidakstabilan sehubungan dengan S

**Syarat:**

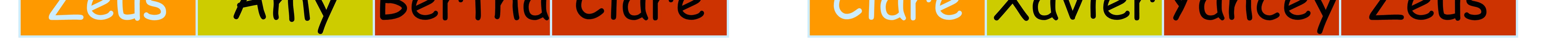
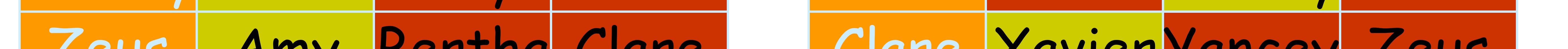
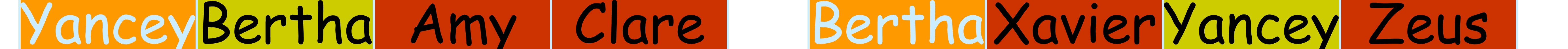
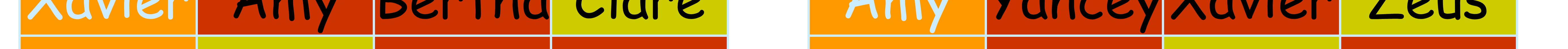
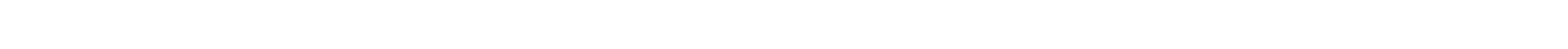
* Perfect Match: semua orang dicocokkan secara monogami.
  1. Setiap pria mendapatkan satu wanita.
  2. Setiap wanita mendapatkan satu pria.
* Stable Matching: pencocokan sempurna tanpa pasangan tidak stabil.

**Stable Matching Problemnya:**

Dengan daftar preferensi pria dan wanita, temukan sebuah stable matching jika ada.

**Contoh 1**

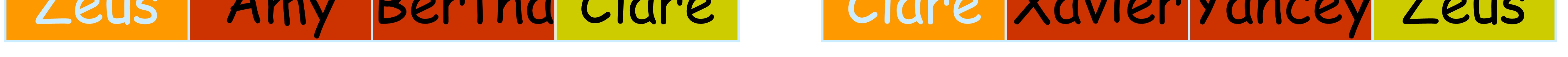
* Pertanyaan: Jika dipasangkan X-C, Y-B, dan Z-A, apakah stabil?



Jawaban: Tidak. Bertha & Xavier akan putus

**Contoh 2**

* Pertanyaan: Jika dipasangkan X-A, Y-B, dan Z-C, apakah stabil?



Jawaban: Ya

# WORKSHEET 1

Jika Anda belum mengerajakan worksheet 01 di kelas, maka Anda dapat mengerjakannya di awal praktikum. Anda diberikan waktu 30 menit untuk menyelesaikan persoalan pada worksheet 01. Bagi Anda yang sudah mengerjakan, Anda dapat langsung mengerjakan tugas praktikum dan mencocokkan hasil worksheet 01 Anda dengan tugas praktikum.

**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.

Men’s Preferences Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0th | 1st | 2nd | 3rd | 4th |
| Bertha | Amy | Diane | Erika | Clare |
| Diane | Bertha | Amy | Clare | Erika |
| Bertha | Erika | Clare | Diane | Amy |
| Amy | Diane | Clare | Bertha | Erika |
| Bertha | Diane | Amy | Erika | Clare |

Victor

Wyatt

Xavier

Yancey

Zeus

Women’s Preferences Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0th | 1st | 2nd | 3rd | 4th |
| Zeus | Victor | Wyatt | Yancey | Xavier |
| Xavier | Wyatt | Yancey | Victor | Zeus |
| Wyatt | Xavier | Yancey | Zeus | Victor |
| Victor | Zeus | Yancey | Xavier | Wyatt |
| Yancey | Wyatt | Zeus | Xavier | Victor |

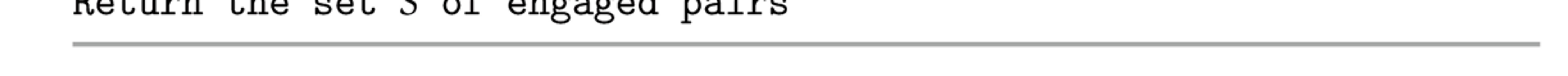
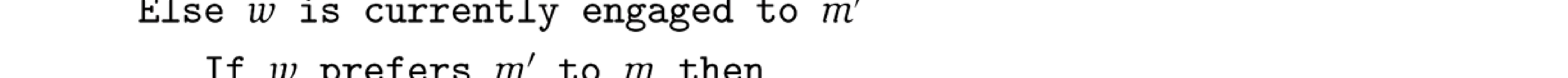
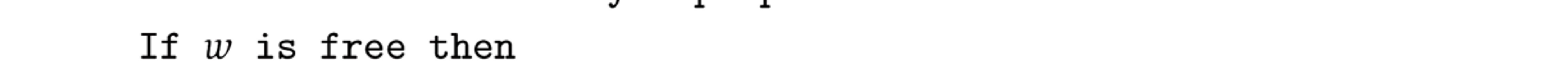
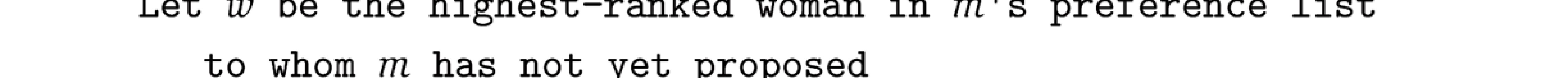
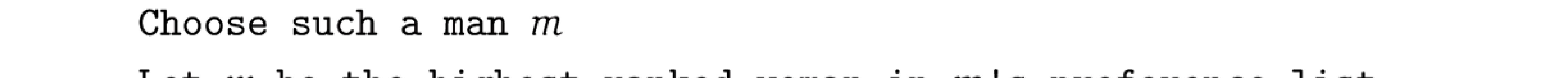
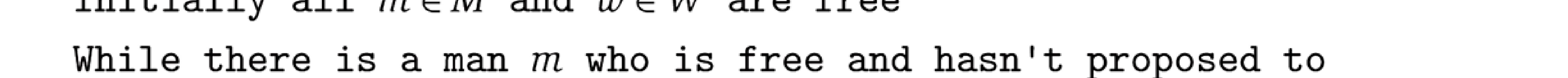
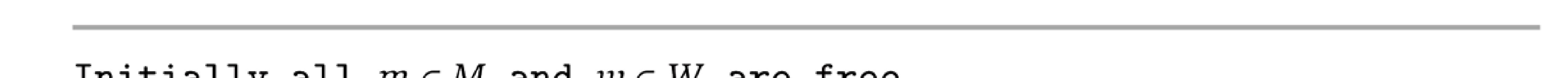
Amy

Bertha

Clare

Diane

Erika



JAWABAN WORKSHEET 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Victor | Bertha | Amy |  |  |  |
| Wyatt | Diane | Bertha | Amy | Clare |  |
| Xavier | Bertha |  |  |  |  |
| Yancey | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane |  |  |  |

Victor -> Amy

Wyatt -> Clare

Xavier -> Bertha

Yancey -> Erika

Zeus -> Diane

# 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.

JAWABAN TUGAS PRAKTIKUM

// C++ program for stable marriage problem

#include <iostream>

#include <string.h>

#include <stdio.h>

using namespace std;

// Number of Men or Women

#define N 5

// This function returns true if woman 'w' prefers man 'm1' over man 'm'

bool wPrefersM1OverM(int prefer[2\*N][N], int w, int m, int m1)

{

// Check if w prefers m over her current engagment m1

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

{

// If m1 comes before m in lisr of w, then w prefers her

// cirrent engagement, don't do anything

if (prefer[w][i] == m1)

return true;

// If m cmes before m1 in w's list, then free her current

// engagement and engage her with m

if (prefer[w][i] == m)

return false;

}

}

// Prints stable matching for N boys and N girls. Boys are numbered as 0 to

// N-1. Girls are numbereed as N to 2N-1.

void stableMarriage(int prefer[2\*N][N])

{

// Stores partner of women. This is our output array that

// stores paing information. The value of wPartner[i]

// indicates the partner assigned to woman N+i. Note that

// the woman numbers between N and 2\*N-1. The value -1

// indicates that (N+i)'th woman is free

int wPartner[N];

// An array to store availability of men. If mFree[i] is

// false, then man 'i' is free, otherwise engaged.

bool mFree[N];

// Initialize all men and women as free

memset(wPartner, -1, sizeof(wPartner));

memset(mFree, false, sizeof(mFree));

int freeCount = N;

// While there are free men

while (freeCount > 0)

{

// Pick the first free man (we could pick any)

int m;

for (m = 0; m < N; m++)

if (mFree[m] == false)

break;

// One by one go to all women according to m's preferences.

// Here m is the picked free man

for (int i = 0; i < N && mFree[m] == false; i++)

{

int w = prefer[m][i];

// The woman of preference is free, w and m become

// partners (Note that the partnership maybe changed

// later). So we can say they are engaged not married

if (wPartner[w-N] == -1)

{

wPartner[w-N] = m;

mFree[m] = true;

freeCount--;

}

else // If w is not free

{

// Find current engagement of w

int m1 = wPartner[w-N];

// If w prefers m over her current engagement m1,

// then break the engagement between w and m1 and

// engage m with w.

if (wPrefersM1OverM(prefer, w, m, m1) == false)

{

wPartner[w-N] = m;

mFree[m] = true;

mFree[m1] = false;

}

} // End of Else

} // End of the for loop that goes to all women in m's list

} // End of main while loop

// Print the solution

cout << " Wanita \tPria" << endl;

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

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

cout << " " ;

if(i==0){

cout<<"Amy";

}else if(i==1){

cout<<"Bertha";

}else if(i==2){

cout<<"Clare";

}else if(i==3){

cout<<"Diane";

}else{

cout<<"Erika";

}

cout<<"\t\t";

if(wPartner[i]==0){

cout<<"Victor";

}else if(wPartner[i]==1){

cout<<"Wyatt";

}else if(wPartner[i]==2){

cout<<"Xavier";

}else if(wPartner[i]==3){

cout<<"Yancey";

}else{

cout<<"Zeus";

}

cout<<endl;

}

}

// Driver program to test above functions

int main()

{

int prefer[2\*N][N] = {

{6, 5, 8, 9, 7},

{8, 6, 5, 7, 9},

{6, 9, 7, 8, 5},

{5, 8, 7, 6, 9},

{6, 8, 5, 9, 7},

{4, 0, 1, 3, 2},

{2, 1, 3, 0, 4},

{1, 2, 3, 4, 0},

{0, 4, 3, 2, 1},

{3, 1, 4, 2, 0},

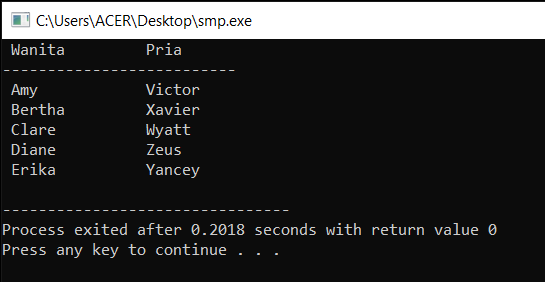
};

stableMarriage(prefer);

return 0;

}

OUTPUT



# ANALISIS ALGORITMA

Jawablah pertanyaan berikut:

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

|  |
| --- |
| Sama, hasil jawaban di worksheet 01 dengan program sama persis. |

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!

|  |
| --- |
| Dalam setiap iterasi pengulangan loop, seorang pria lajang melamar wanita berikutnya dalam daftar pilihannya, seseorang yang belum pernah ia ajukan sebelumnya. Karena ada n laki-laki dan setiap daftar preferensi memiliki panjang n, ada paling banyak proposal n2 yang dapat terjadi. Jadi 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!

|  |
| --- |
| Matching dilakukan dengan syarat perfect matching yang berarti:  Perfect Match: semua orang dicocokkan secara monogami.   * 1. Setiap pria mendapatkan satu wanita.   2. Setiap wanita mendapatkan satu pria.   Artinya, ketika ada seorang pria bebas belum memiliki pasangan. Maka akan ada wanita yang belum diajak bertunangan.  Sebagai contoh: Terdapat pria M dan M’, serta wanita W dan W’. Jika M memilih W, dan M’ bersifat bebas. Maka W’ belum diajak bertunangan. |

**Teorema (1.5):**

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

|  |
| --- |
| Perfect Matching berarti semua orang dicocokkan secara monogami.   1. Setiap pria mendapatkan satu wanita. 2. Setiap wanita mendapatkan satu pria.   Dalam proses eksekusi, setiap pria terbukti selalu mendapatkan satu wanita, dan setiap wanita selalu mendapatkan satu pria.  Sebagai contoh: Worksheet 1. |

**Teorema (1.6):**

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

Buktikan!

|  |
| --- |
| Stable Matching berarti pencocokan sempurna tanpa pasangan tidak stabil.  Syarat Stable Matching adalah:   * Perfect (1 laki-laki tepat berhubungan dengan satu perempuan), dan * tidak ada ketidakstabilan .     Algoritma G-S merupakan algoritma yang bertujuan untuk memasangkan dua himpunan sama panjang menjadi sebuah hubungan set pasangan yang stabil.  Worksheet 1 menjadi contoh sekaligus bukti bahwa Algoritma G-S memberikan hasil set pasangan stabil di akhir eksekusinya. |