

LAPORAN PRAKTIKUM 1  
Analysis Algorithm



Penyusun :  
Mohammad Dhikri  
140810180075

PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN  
2020

**Worksheet 01**

Jika Anda belum mengerjakan 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 preference-lists berikut ini. Gunakan processor terhebat yang Anda miliki (otak) untuk mengikuti algoritma G-S dan output tidak perlu diuraikan per-looping tetapi Anda harus memahami hasil setiap looping.

Men's Preferences Profile					
	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
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

Women's Preferences Profile					
	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
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

## Algoritma Gale - Shapley

---

```
Initially all  $m \in M$  and  $w \in W$  are free
While there is a man  $m$  who is free and hasn't proposed to
every woman
    Choose such a man  $m$ 
    Let  $w$  be the highest-ranked woman in  $m$ 's preference list
    to whom  $m$  has not yet proposed
    If  $w$  is free then
         $(m, w)$  become engaged
    Else  $w$  is currently engaged to  $m'$ 
        If  $w$  prefers  $m'$  to  $m$  then
             $m$  remains free
        Else  $w$  prefers  $m$  to  $m'$ 
             $(m, w)$  become engaged
             $m'$  becomes free
        Endif
    Endif
Endwhile
Return the set  $S$  of engaged pairs
```

---

1. Victor -> Bertha (Engaged)

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor	Bertha				
Wyatt					
Xavier					
Yancey					
Zeus					

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy					
Bertha				Victor	

Clare					
Diane					
Erika					

2. Wyatt -> Diane (Engaged)

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor	Bertha				
Wyatt	Diane				
Xavier					
Yancey					
Zeus					

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy					
Bertha				Victor	
Clare					
Diane					Wyatt
Erika					

3. Xavier -> Bertha (Engaged)      Victor -> Free

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor					
Wyatt	Diane				
Xavier	Bertha				
Yancey					
Zeus					

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy					
Bertha	Xavier				
Clare					
Diane					Wyatt
Erika					

4. Yancey -> Amy (Engaged)

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor					
Wyatt	Diane				

Xavier	Bertha				
Yancey	Amy				
Zeus					

#### Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy				Yancey	
Bertha	Xavier				
Clare					
Diane					Wyatt
Erika					

5. Zeus -> Bertha (X)      Zeus -> Free

#### Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor					
Wyatt	Diane				
Xavier	Bertha				
Yancey	Amy				
Zeus					

#### Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy				Yancey	
Bertha	Xavier				
Clare					
Diane					Wyatt
Erika					

6. Victor -> Amy (Engaged)      Yancey -> Free

#### Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor		Amy			
Wyatt	Diane				
Xavier	Bertha				
Yancey					
Zeus					

#### Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy		Victor			
Bertha	Xavier				

Clare					
Diane					Wyatt
Erika					

7. Zeus -> Diane (Engaged)      Wyatt -> Free

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor		Amy			
Wyatt					
Xavier	Bertha				
Yancey					
Zeus		Diane			

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy		Victor			
Bertha	Xavier				
Clare					
Diane		Zeus			
Erika					

8. Yancey -> Clare (Engaged)

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor		Amy			
Wyatt					
Xavier	Bertha				
Yancey			Clare		
Zeus		Diane			

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy		Victor			
Bertha	Xavier				
Clare			Yancey		
Diane		Zeus			
Erika					

9. Wyatt -> Clare (Engaged)      Yancey -> Free

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor		Amy			
Wyatt				Clare	
Xavier	Bertha				
Yancey					
Zeus		Diane			

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy		Victor			
Bertha	Xavier				
Clare	Wyatt				
Diane		Zeus			
Erika					

10. Yancey -> Erika (Engaged)

Men's Preferences Profile

Men's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Victor		Amy			
Wyatt				Clare	
Xavier	Bertha				
Yancey					Erika
Zeus		Diane			

Women's Preferences Profile

Women's Name	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
Amy		Victor			
Bertha	Xavier				
Clare	Wyatt				
Diane		Zeus			
Erika	Yancey				

Hasil :

Victor Engaged Amy  
 Wyatt Engaged Clare  
 Xavier Engaged Bertha  
 Yancey Engaged Erika

## Zeus Engaged Diane

Analisis Algoritma :

```
/*
Name : Mohammad Dhikri
NPM : 140810180075
Kelas : A
*/

#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
#define N 5

bool chooseMen(int priority[2 * N][N], int w, int m, int m1){
    for (int i = 0; i < N; i++)
    {
        if (priority[w][i] == m1)
            return true;

        if (priority[w][i] == m)
            return false;
    }
}

void printPeer(int priority[2 * N][N]) {
    bool singleMan[N];
    int peerWoman[N];
    memset(peerWoman, -1, sizeof(peerWoman));
    memset(singleMan, false, sizeof(singleMan));
    int jumlah = N;
    while (jumlah > 0){
        int m;
        for (m = 0; m < N; m++)
            if (singleMan[m] == false)
                break;

        for (int i = 0; i < N && singleMan[m] == false; i++){
            int w = priority[m][i];
            if (peerWoman[w - N] == -1){
                peerWoman[w - N] = m;
                singleMan[m] = true;
            }
        }
        jumlah--;
    }
}
```



```

        jumlah--;
    }
    else{
        int m1 = peerWoman[w - N];
        if (chooseMen(priority, w, m, m1) == false){
            peerWoman[w - N] = m;
            singleMan[m] = true;
            singleMan[m1] = false;
        }
    }
}
}
}
cout<<"Pasangan yang stabil : "<<endl<<endl;
string man;
string woman;
for (int i = 0; i < N; i++){
    if (i < N){
        if (peerWoman[i] == 0)
            man = "Victor";
        if (peerWoman[i] == 1)
            man = "Wyatt";
        if (peerWoman[i] == 2)
            man = "Xavier";
        if (peerWoman[i] == 3)
            man = "Yancey";
        if (peerWoman[i] == 4)
            man = "Zeus";

        if (i == 0)
            woman = "Amy";
        if (i == 1)
            woman = "Bertha";
        if (i == 2)
            woman = "Clare";
        if (i == 3)
            woman = "Diane";
        if (i == 4)
            woman = "Erika";
    }
    cout << " " << man << "\t " << woman << endl;
}
}
int main(){
    int priority[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},
        {3, 1, 4, 2, 0}};
printPeer(priority);
return 0;
}

```

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

Alhamdulillah jawabannya sama

2. 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).  
→ Terpenuhi
- Fakta (1.2): Urutan wanita yang dilamar pria lebih buruk dan lebih buruk lagi (hal ini sesuai dengan daftar preferensi pria).  
→ Terpenuhi
- Teorema (1.3): Algoritma G-S berakhir setelah paling banyak  $n^2$  iterasi menggunakan While Loop. Buktikan!  
→ Jumlah iterasi yang dapat terjadi paling banyak adalah  $n^2$  karena sebanyak  $n$  pria akan melamar minimal sekali  $n$  preferensi wanita sampai terjadi kecocokan.
- 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 jumlah pasangan yang ada. Misal, ada  $n$  wanita yang bertunangan dan ada  $n$  pria yang bertunangan, yang berarti bahwa 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 wanita 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 urutan 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!

→ Setiap pria pasti pada akhirnya berpasangan. 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. Selain itu pasangan yang sudah dipasangkan juga harus menunjukkan kecocokan dengan preferensi masing-masing dan pencocokan algoritma G-S dianggap selalu stabil.

