

# **LAPORAN PRAKTIKUM I**

## **ANALISIS ALGORITMA**



**Disusun Oleh :**  
**Cryssa Aprilia Ermiyanda Piter**  
**140810170063**

**FAKULTAS MATEMATIKA DAN ILMU ALAM**  
**PROGRAM STUDI TEKNIK INFORMATIKA**  
**UNIVERSITAS PADJADJARAN**  
**2019**

## 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

---

```

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

```

---

Langkah-langkah :

1. ~~Victor~~ — ~~Bertha~~
2. ~~Wyatt~~ — ~~Diane~~
3. **Xavier** — **Bertha**

**Victor (Lajang)**

4. Victor – Amy

5. Yancey – ~~Amy~~

**Yancey (Lajang)**

6. Yancey – ~~Diane~~

**Wyatt (Lajang)**

7. Wyatt – ~~Bertha~~

**Wyatt (Lajang)**

8. Wyatt – ~~Amy~~

**Wyatt (Lajang)**

9. Wyatt – Clare

10. Zeus – ~~Bertha~~

**Zeus (Lajang)**

11. Zeus – ~~Diane~~

**Yancey (Lajang)**

12. Yancey – ~~Clare~~

**Yancey (Lajang)**

13. Yancey – ~~Bertha~~

**Yancey (Lajang)**

14. Yancey – Erika

## 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 analisislah (Poin ini disampaikan pada bagian Analisis Algoritma) yang sudah disiapkan.

## Analisis Algoritma

Jawablah pertanyaan berikut:

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

Iya, sama.

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  $n^2$  iterasi menggunakan While Loop. Buktikan!

Dalam setiap iterasi pengulangan loop, seorang pria lajang melamar wanita berikutnya yang ada dalam daftar pilihannya, yaitu wanita yang belum ia lamar sebelumnya. Karena ada  $n$  laki-laki dan daftar pilihan memiliki panjang  $n$ , ada paling banyak  $n^2$  lamaran yang dapat terjadi. Jadi, jumlah iterasi paling banyak yang dapat terjadi adalah  $n^2$ .

Buktinya, pencocokan yang dikembalikan menghasilkan pasangan yang stabil.

**Teorema (1.4):**

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

Buktikan!

Buktinya berdasarkan kontradiksi. Misalkan ada waktu tertentu dalam pelaksanaan algoritma ketika seorang pria lajang, namun telah mengusulkan kepada setiap wanita. Ini berarti, setiap wanita telah diusulkan setidaknya satu kali. Dengan teori 1, mendapatkan bahwa setiap wanita bertunangan. Jadi, kita telah melibatkan  $n$  wanita dan karenanya  $n$  laki-laki bertunangan, yang menyiratkan bahwa  $m$  juga terlibat bertunangan dengan asumsi bahwa  $m$  adalah lajang.

**Teorema (1.5):**

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

Hal ini disebabkan karena semua laki-laki dan perempuan saling berpasangan.

**Teorema (1.6):**

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

Buktikan!

Menunjukkan bahwa pencocokan yang dikembalikan adalah pencocokan sempurna. Buktinya dengan kontradiksi. Misalkan tidak, maka ada seorang pria yang masih lajang di akhir algoritma. Menurut teori 2, itu berarti  $m$  belum melamar beberapa wanita. Tetapi kemudian, algoritma tidak akan keluar dari pengulangan loop, menghasilkan kontradiksi yang diinginkan.

Menunjukkan bahwa pencocokan yang dikembalikan stabil. Lagi-lagi buktinya dengan kontradiksi. Misalkan ada laki-laki  $m$  dan  $m'$  dan wanita  $w$  dan  $w'$  sehingga  $(m, w)$  dan  $(m', w')$  berada di  $S$ , tetapi  $m$  lebih suka  $w'$  ke  $w$  dan  $w'$  lebih suka  $m$  ke  $m'$ . Dengan algoritma,  $w$  adalah wanita terakhir yang saya ajukan. Karena  $m$  lebih suka  $w'$  ke  $w$ ,  $m$  harus sudah mengusulkan ke  $w'$  sebelum usulannya ke  $w$ . Pada saat itu, atau nanti,  $w'$  bertunangan dengan seorang pria, katakanlah  $m''$ , yang ia sukai lebih dari  $m$ . Pada akhirnya,  $w'$  bertunangan dengan  $m'$ . Oleh teori 1, menemukan bahwa  $w'$  lebih memilih  $m'$  daripada  $m''$  dan lebih memilih  $m''$  daripada  $m$ ; ini menyiratkan bahwa  $w'$  lebih suka  $m'$  daripada  $m$ , bertentangan dengan asumsi bahwa  $w'$  lebih memilih  $m$  daripada  $m'$ .

