# LAPORAN PRAKTIKUM 1
# Analisis algoritma


**Archi Cantona Rusanggara**
**140810180050**

## 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 perusahaan- setelah 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?

(i) E lebih memilih setiap satu dari daftar pelamar yang diterima(A); atau
(ii) 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 $M=\{N_1, \ldots, N_n\}$ dari n pria, dan satu set $W=\{w_1, \ldots, w_n\}$ dari n wanita. Produk kartesius M x W menunjukkan set dari semua pasangan bentuk yang mungkin dipesan (N,w), di mana $N \in M$ dan $w \in W$.

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

Dipandu oleh motivasi awal kita dalam hal pemberi kerja dan pelamar, kita harus khawatir tentang situasi berikut: Ada dua pasangan (N, w) dan (N′, w′) dalam S (seperti yang digambarkan pada Gambar 1.1) dengan properti bahwa N lebih suka w' daripada w, dan w′ lebih suka N ke N′. Dalam hal ini, tidak ada yang bisa menghentikan N dan w′ 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

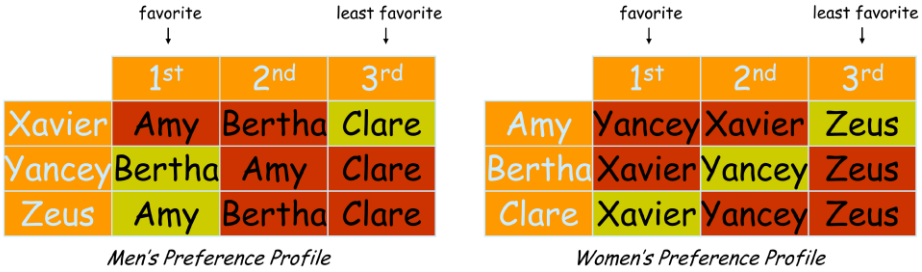
**Syarat:**

- 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**

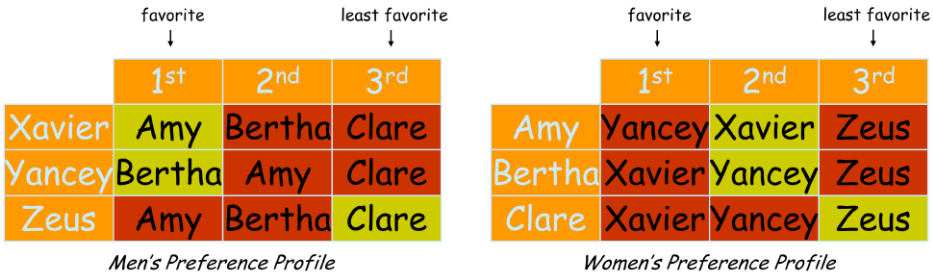
- Pertanyaan: Jika dipasangkan X-C, Y-B, dan Z-A, apakah stabil?




- Jawaban: Tidak. Bertha & Xavier akan putus

**Contoh 2**

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




- Jawaban: Ya

## Worksheet 01

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 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

| | 0th | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|---|
| 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

| | 0th | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|---|
| Amy | 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 |

## 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.

| Tahap | Man | Woman | Free |
|---|---|---|---|
| 1 | Victor | Bertha | |
| 2 | Wyatt | Diane | |
| 3 | Xavier | Bertha | Victor |
| 4 | Victor | Amy | |
| 5 | Yancey | Amy | Yancey |
| 6 | Yancey | Diane | Wyatt |
| 7 | Wyatt | Bertha | Wyatt |
| 8 | Wyatt | Amy | Wyatt |
| 9 | Wyatt | Clare | |
| 10 | Zeus | Bertha | Zeus |
| 11 | Zeus | Diane | Yancey |
| 12 | Yancey | Clare | Yancey |
| 13 | Yancey | Bertha | Yancey |
| 14 | Yancey | Erika | |

## Analisis Algoritma

Jawablah pertanyaan berikut:

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

> Jawaban saya sama dengan apa yang saya kerjakan saat dikelas

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!

> Pada saat menggunakan while loop memiliki kemajuan , yaitu dimana pria lajang melamar wanita berikutnya dalam daftar pilihannya, lalu seseorang yang belum pernah ia ajukan sebelumnya. Karena ada n laki-laki dan setiap daftar prefensi memiliki panjang n, ada paling banyak proposal n2 ynag dapat terjadi. Jadi jumlah dalam iterasi yang bisa 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!

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 bertentangan dengan asumsi bahwa m adalah lajang.

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

Karena setiap laki laki mendapatkan pasangan wanita

**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'.

```cpp
#include<iostream>
    using namespace std;

    typedef int prefer[5];

    struct orang{
        string nama;
        int pasangan;
        int preferences[5];
    };

    orang setDataOrang(string nama, prefer prefers);
    void stableMatching(orang pria[5], orang wanita[5]);

    int main(){
        orang pria[5], wanita[5];
        prefer prefers;

        //Pria
        prefers[0] = 1; prefers[1] = 0; prefers[2] = 3; prefers[3] = 4; prefers[4] = 2;
        pria[0] = setDataOrang("Victor",prefers);
        prefers[0] = 3; prefers[1] = 1; prefers[2] = 0; prefers[3] = 2; prefers[4] = 4;
        pria[1] = setDataOrang("Wyatt",prefers);
        prefers[0] = 1; prefers[1] = 4; prefers[2] = 2; prefers[3] = 3; prefers[4] = 0;
        pria[2] = setDataOrang("Xavier",prefers);
        prefers[0] = 0; prefers[1] = 3; prefers[2] = 2; prefers[3] = 1; prefers[4] = 4;
        pria[3] = setDataOrang("Yancey",prefers);
        prefers[0] = 1; prefers[1] = 3; prefers[2] = 0; prefers[3] = 4; prefers[4] = 2;
        pria[4] = setDataOrang("Zeus",prefers);

        //Wanita
        prefers[0] = 4; prefers[1] = 0; prefers[2] = 1; prefers[3] = 3; prefers[4] = 2;
        wanita[0] = setDataOrang("Amy",prefers);
        prefers[0] = 2; prefers[1] = 1; prefers[2] = 3; prefers[3] = 0; prefers[4] = 4;
        wanita[1] = setDataOrang("Bertha",prefers);
        prefers[0] = 1; prefers[1] = 2; prefers[2] = 3; prefers[3] = 4; prefers[4] = 0;
        wanita[2] = setDataOrang("Clare",prefers);
        prefers[0] = 0; prefers[1] = 4; prefers[2] = 3; prefers[3] = 2; prefers[4] = 1;
        wanita[3] = setDataOrang("Diane",prefers);
        prefers[0] = 3; prefers[1] = 1; prefers[2] = 4; prefers[3] = 2; prefers[4] = 0;
        wanita[4] = setDataOrang("Erika",prefers);

        stableMatching(pria,wanita);

        cout << "\nHasil Stable Matching :" << endl;
        cout << "-----------------" << endl
             << "| Man \t Woman\t|" << endl
```

```cpp
        stableMatching(pria,wanita);

        cout << "\nHasil Stable Matching :" << endl;
        cout << "-----------------" << endl
             << "| Man \t Woman\t|" << endl
             << "-----------------" << endl;
        for(int i = 0; i < 5; i++){
            cout << "|" << pria[i].nama << "\t " << wanita[pria[i].pasangan].nama << "\t|" << endl;
        }
        cout << "-----------------" << endl;
    }

    orang setDataOrang(string nama, prefer prefers){
        orang org;
        org.nama = nama;
        org.pasangan = -1;
        for(int i = 0; i < 5;i++){
            org.preferences[i] = prefers[i];
        }
        return org;
    }

    void stableMatching(orang pria[5], orang wanita[5]){
        bool bebas = true;
        int i=0, j=0, prefSek,prefSuk;
        while(bebas){
            if(wanita[pria[i].preferences[j]].pasangan = -1){
                pria[i].pasangan = pria[i].preferences[j];
                wanita[pria[i].preferences[j]].pasangan = i;
            }else{
              int k = 0;
              prefSek = -1; prefSuk = -1;
              while(prefSek == -1 || prefSuk == -1){
                  if(i == wanita[pria[i].preferences[j]].preferences[k]){
                      prefSuk = k;
                  }
                  if(wanita[pria[i].preferences[j]].pasangan == wanita[pria[i].preferences[j]].preferences[k]){
                      prefSek = k;
                  }
                  k++;
              }
              if(prefSuk<prefSek){
                  pria[wanita[pria[i].preferences[j]].pasangan].pasangan = -1;
                  pria[i].pasangan = pria[i].preferences[j];
                  wanita[pria[i].preferences[j]].pasangan = i;
              }else{
                  j++;
                  continue;
```

```
                    }
                    k++;
                }
                if(prefSuk<prefSek){
                    pria[wanita[pria[i].preferences[j]].pasangan].pasangan = -1;
                    pria[i].pasangan = pria[i].preferences[j];
                    wanita[pria[i].preferences[j]].pasangan = i;
                }else{
                    j++;
                    continue;
                }
            }
            bebas = false;
            j=0;
            for(int l = 0; l < 5; l++){
                if(pria[l].pasangan == -1){
                    i = l;
                    bebas = true;
                    break;
                }
            }
        }
    }


```

```
Hasil Stable Matching :
-----------------
| Man    Woman  |
-----------------
|Victor  Bertha |
|Wyatt   Diane  |
|Xavier  Bertha |
|Yancey  Amy    |
|Zeus    Bertha |
-----------------

--------------------------------
Process exited after 0.06469 seconds with return value 0
Press any key to continue . . .
```

## Teknik Pengumpulan

- Lakukan push ke github/gitlab untuk semua program dan laporan hasil analisa yang berisi jawaban dari pertanyaan-pertanyaan yang diajukan. Silahkan sepakati dengan asisten praktikum.

## Penutup

- Ingat, berdasarkan Peraturan Rektor No 46 Tahun 2016 tentang Penyelenggaraan Pendidikan, mahasiswa wajib mengikuti praktikum 100%
- Apabila tidak hadir pada salah satu kegiatan praktikum segeralah minta tugas pengganti ke asisten praktikum
- Kurangnya kehadiran Anda di praktikum, memungkinkan nilai praktikum Anda tidak akan dimasukkan ke nilai mata kuliah.