**LAPORAN PRAKTIKUM**

**ANALISIS ALGORITMA**



**DISUSUN OLEH**

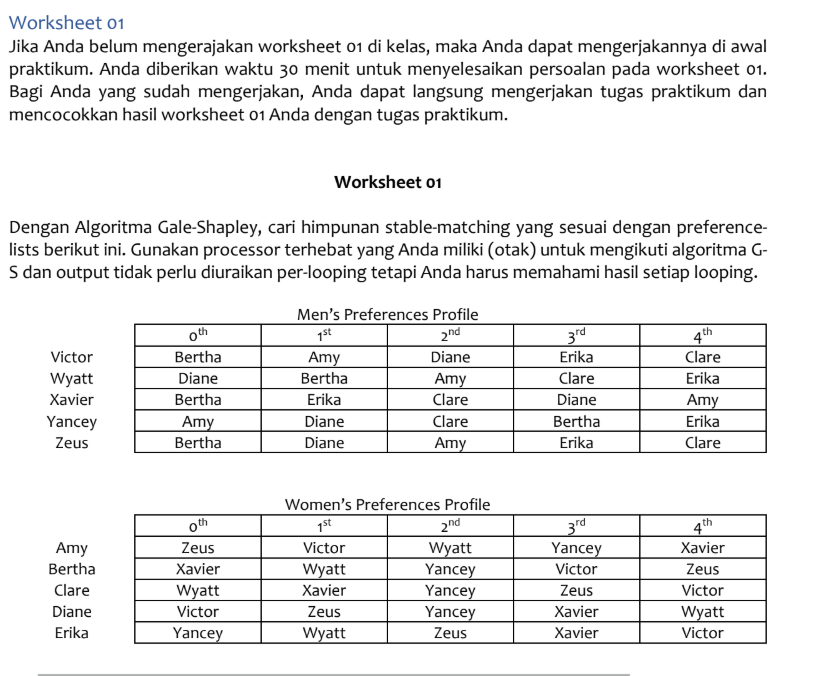
Gede Bagus Darmagita - 140810180068

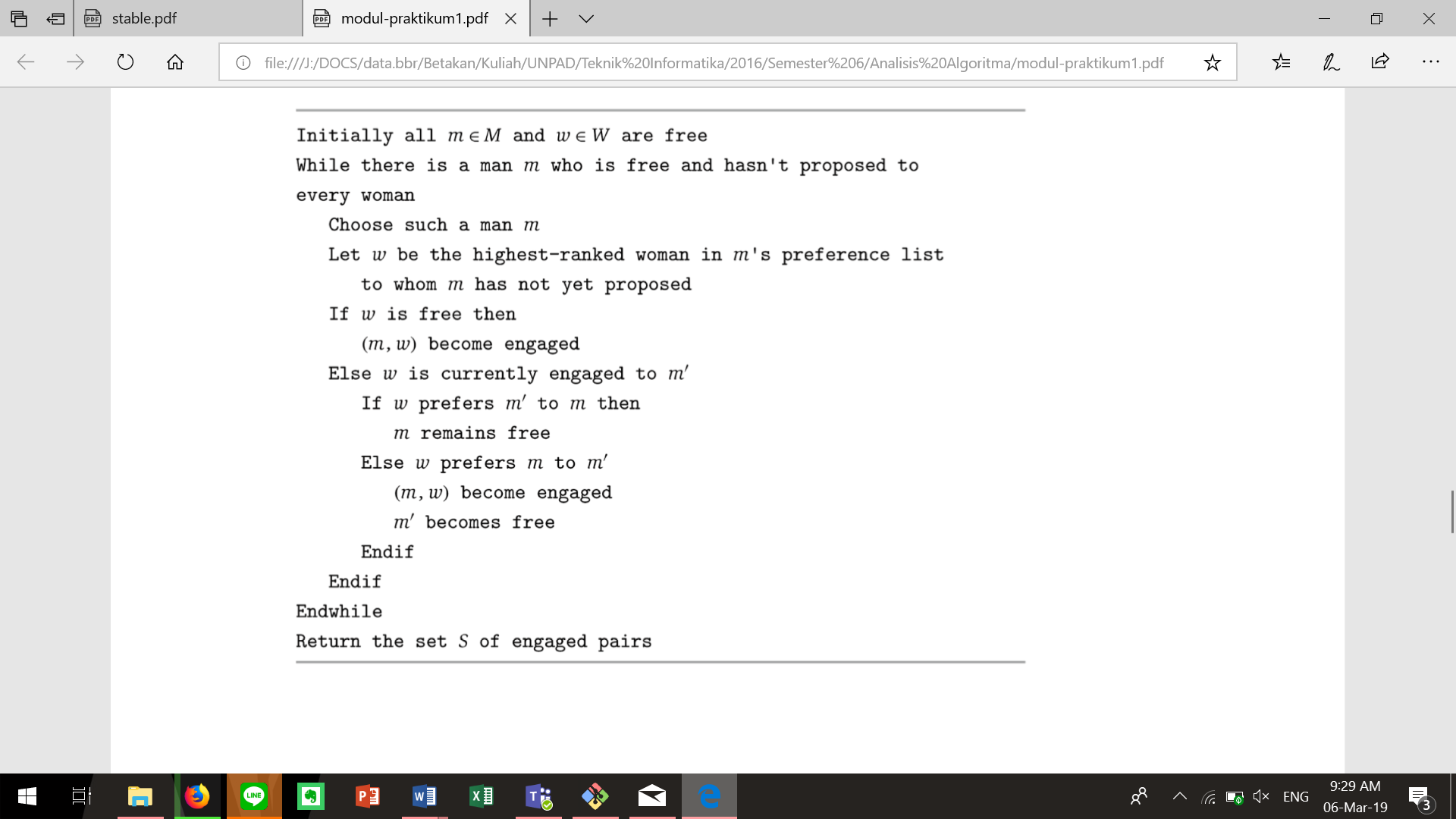
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2020**



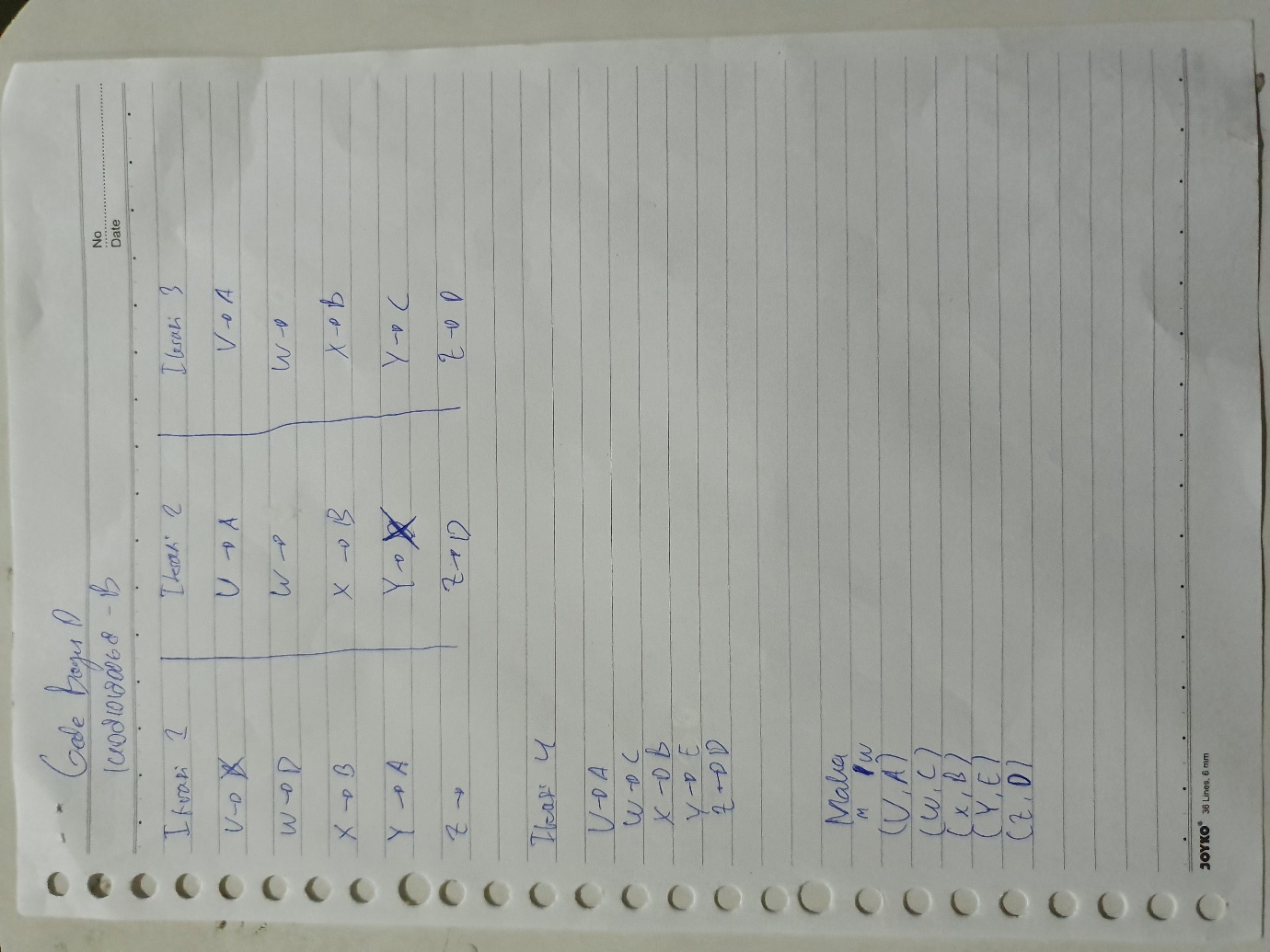


Pertanyaan:

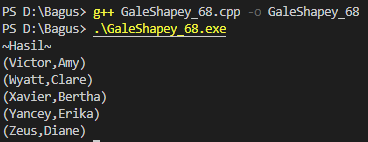
* Ubahlah pseudocode algoritma G-S pada worksheet 01 ke dalam program menggunakan bahasa C++.
* Gunakan table pria sebagai table acuan untuk memudahkan Anda menentukan pasangannya.
* Cocokkan jawaban Anda pada worksheet 01 dengan hasil program yang Anda buat.
* Jika ada yang berbeda tuliskan bagian mana yang berbeda dan analisalah:
  + 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:
    - 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!
    - Teorema (1.4): Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan. Buktikan!
    - Teorema (1.5): Himpunan S yang dikembalikan saat terminasi adalah perfect matching Buktikan!
    - Teorema (1.6): Sebuah eksekusi algoritma G-S mengembalikan satu set pasangan S. Set S adalah pasangan yang stabil. Buktikan!

**Answer**

Tulis:



Program:



Code:

/\*

    Nama    : Gede Bagus Darmagita

    NPM     : 140810180068

    Kelas   : B

\*/

#include <iostream>

#include <string>

using namespace std;

const int N = 5;

string Nama\_Pria[N] = {"Victor", "Wyatt", "Xavier", "Yancey", "Zeus"};

string Nama\_Wanita[N] = {"Amy", "Bertha", "Clare", "Diane", "Erika"};

int Pria\_preference[N][N] = {

    {1, 0, 3, 4, 2},

    {3, 1, 0, 2, 4},

    {1, 4, 2, 3, 0},

    {0, 3, 2, 1, 4},

    {1, 3, 0, 4, 2},

};

int Wanita\_preference[N][N] = {

    {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},

};

int stable[N][2];

bool Pria\_Match[N];

bool Wanita\_match[N];

int getPriaMatch(int wanita)

{

    for (int i = 0; i < N; i++)

    {

        if (stable[i][1] == wanita)

        {

            return stable[i][0];

            break;

        }

    }

    return -1;

}

void matchingProcess(int M, int W)

{

    stable[M][0] = M;

    stable[M][1] = W;

    Pria\_Match[M] = true;

    Wanita\_match[W] = true;

}

int getPreferenceLevel(int wanita, int pria)

{

    for (int i = 0; i < N; i++)

    {

        if (Wanita\_preference[wanita][i] == pria)

        {

            return i;

            break;

        }

    }

    return -1;

}

void traversalMatch()

{

    cout << "~Hasil~\n";

    for (int i = 0; i < N; i++)

    {

        if (i != 0)

        {

            cout << "\n";

        }

        cout << "(" << Nama\_Pria[stable[i][0]] << "," << Nama\_Wanita[stable[i][1]] << ")";

    }

}

int main()

{

    for (int i = 0; i < N; i++)

    {

        for (int j = 0; j < N; j++)

        {

            int M = j;

            if (Pria\_Match[M] == false)

            {

                int W = Pria\_preference[M][i];

                if (Wanita\_match[W] == true)

                {

                    int xM = getPriaMatch(W);

                    int xM\_prep = getPreferenceLevel(W, xM);

                    int M\_prep = getPreferenceLevel(W, M);

                    if (M\_prep < xM\_prep)

                    {

                        matchingProcess(M, W);

                        Pria\_Match[xM] = false;

                        stable[xM][1] = -1;

                        continue;

                    }

                    else

                    {

                        continue;

                    }

                }

                else

                {

                    matchingProcess(M, W);

                    continue;

                }

            }

            else

            {

                continue;

            }

        }

    }

    traversalMatch();

}

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

Jawab: Sama Persis

1. 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!

Jawab:

Karena pada pool terdapat n pria dan n wanita, masing masing pria memiliki preferensi sebanyak n maka jumlah iterasi maksimal yang dapat terjadi adalah . Lalu algoritma ini terdiri dari pria yang melamar wanita yang belum dilamar.

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

Ketika seorang pria sudah mendapatkan pasangan maka dia tidak dapat melamar wanita lain sampai wanita tersebut dilamar oleh pria lain yang preferensinya lebih tinggi dari dia. Lalu wanita akan berganti pasangan dan pria tersebut akan mencari pasanganya pada iterasi berikutnya hingga semua pasangan akan dipastikan stabil.

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

Algoritma ini akan terus berjalan ketika masih ada pria yang masih single dan belum melamar semua wanita sesuai preferensinya hingga setiap pria dan wanita memilik pasangan masing masing, oleh karena itu Himpunan S adalah perfect matching.

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

Karena pada setiap iterasi akan di prioritaskan preferensi dari pihak pria, dan wanita tidak dapat menolak lamaran dari pria apa bila belum memiliki pasangan, maka akan dihasilakan satu set pasangan matching, dan juga meskipun wanita yang di lamar oleh pria lain yang preferensinya lebih tinggi daripada pasangan yang sekarang maka wanita dapat mengganti pasangan bersama pria yang baru melamar tersebut. Namun jika pria tidak bisa mendapatkan wanita tersebut dia akan melamar wanita lainnya pada iterasi berikutnya, sehingga pasti setiap eksekusi akan menghasilkan satu buah pasangan.