**LAPORAN TUGAS BESAR 2**

**IF2211 Strategi Algoritma**

**Simulasi Penyebaran Virus Penyakit dengan Memanfaatkan Algoritma BFS untuk Penelusuran pada Graf**



Disusun oleh:

Kelompok CirengAntiCorona

* Hengky Surya Angkasa 13518048
* Naufal Dean Anugrah 13518123
* Gregorius Jovan Kresnadi 13518135

**Prodi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2020**

# DAFTAR ISI

[**DAFTAR ISI**](#_3bb5zre14oox) **2**

[**BAB I**](#_gy3jq5nhz63r) **:** [**DESKRIPSI TUGAS**](#_yr3dr6x2ddpl) **3**

[**BAB II :**](#_5a336l2r402s) [**DASAR TEORI**](#_m0v9o3nw3n0w) **10**

2.1 [Algoritma Breadth First Search](#_2p9ydub7xuvi) 10

[2.2 Graf](#_2vkh1ugls155) 10

[**BAB III :**](#_q599r0qqa1c2) [**ANALISIS PEMECAHAN MASALAH**](#_hfpktehmdm0h) **11**

[3.1 Struktur Data dan Spesifikasi Program](#_jhmtt7cptovn) 11

3.2 L[angkah-Langkah Pemecahan Masalah](#_yyns9spvgj8p) 11

[**BAB IV**](#_ssxfo83yta14) **:** [**IMPLEMENTASI DAN PENGUJIAN**](#_3z7xyx5y9b8t) **14**

4.1 I[mplementasi Program](#_hitb69o9kdb9) 14

4.2 P[engujian](#_ln6dhi41qphv) 14

[**BAB V**](#_m5rbx7h33y21) [**: KESIMPULAN DAN SARAN**](#_qdjss17u2l2n) **17**

[5.1 Kesimpulan](#_31fqaumlnj20) 17

5.2 S[aran](#_fbyioyf6kt5b) 17

[**DAFTAR PUSTAKA**](#_xkrvrngf6cy4) **18**

# 

# BAB I

# DESKRIPSI TUGAS

Di awal tahun 2020 ini, dunia digemparkan dengan munculnya suatu virus Corona jenis baru yang diberi nama *COVID-19*. Virus mewabah pertama kali di Wuhan, China, sejak awal tahun 2020 lalu telah menginfeksi lebih dari 69.000 orang di seluruh dunia hingga saat ini. Mudahnya penyebaran antarmanusia menyebabkan virus dengan mudahnya tersebar ke seluruh penjuru dunia mulai dari Asia Tenggara, Eropa, Australia, dan Amerika Serikat.

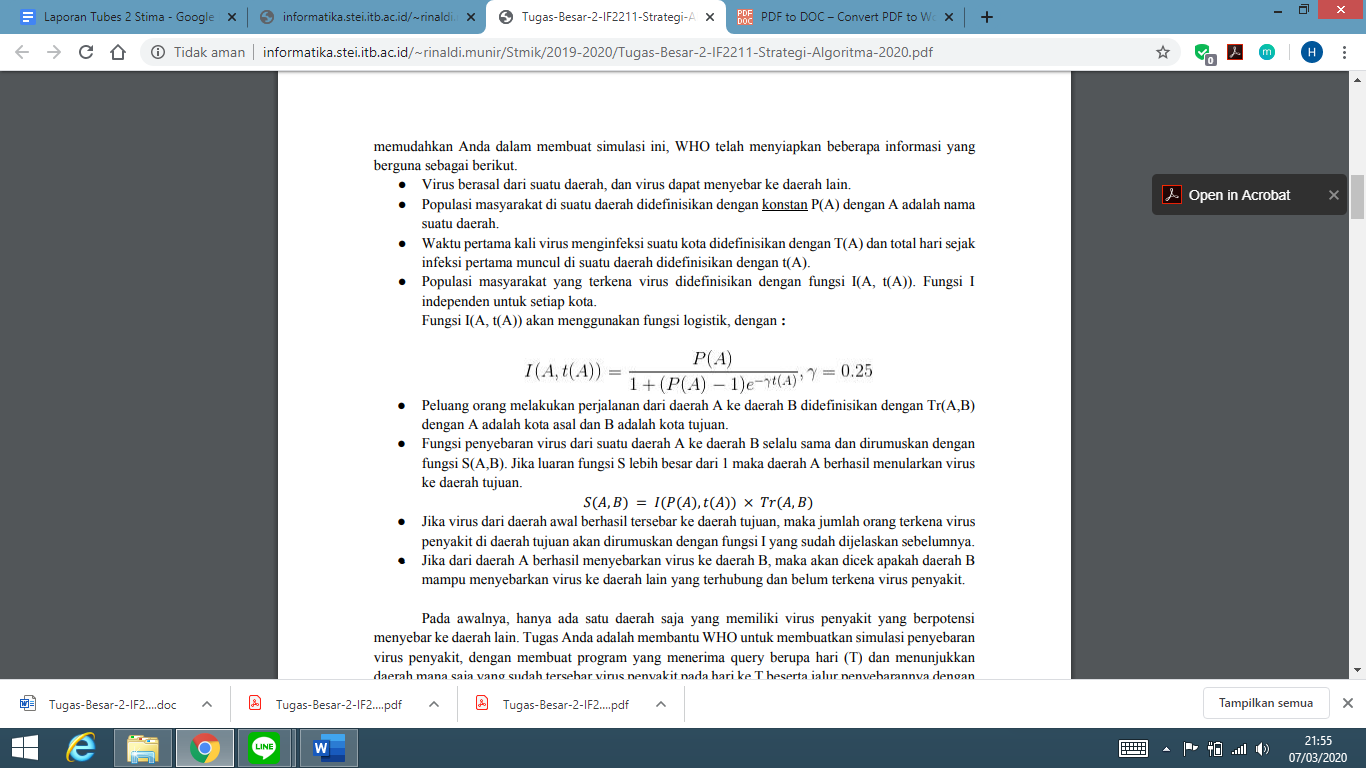
Melihat potensi ancaman yang diakibatkan oleh penyebaran virus penyakit ini, badan kesehatan dunia (WHO) meminta Anda sebagai seorang akademisi di bidang informatika untuk melakukan simulasi penyebaran virus penyakit yang memanfaatkan algoritma BFS. Suatu daerah dapat dianggap sebagai sebuah simpul pada graf, dan hubungan antara dua daerah akan membentuk sisi yang memiliki arah dengan bobot berupa peluang orang melakukan perjalanan antar daerah tersebut. Untuk memudahkan Anda dalam membuat simulasi ini, WHO telah menyiapkan beberapa informasi yang berguna sebagai berikut.

* Virus berasal dari suatu daerah, dan virus dapat menyebar ke daerah lain.
* Populasi masyarakat di suatu daerah didefinisikan dengan konstan P(A) dengan A adalah nama suatu daerah.
* Waktu pertama kali virus menginfeksi suatu kota didefinisikan dengan T(A) dan total hari sejak infeksi pertama muncul di suatu daerah didefinisikan dengan t(A).
* Populasi masyarakat yang terkena virus didefinisikan dengan fungsi I(A, t(A)). Fungsi I independen untuk setiap kota.

Fungsi I(A, t(A)) akan menggunakan fungsi logistik, dengan **:**

****

* Peluang orang melakukan perjalanan dari daerah A ke daerah B didefinisikan dengan Tr(A,B) dengan A adalah kota asal dan B adalah kota tujuan.
* Fungsi penyebaran virus dari suatu daerah A ke daerah B selalu sama dan dirumuskan dengan fungsi S(A,B). Jika luaran fungsi S lebih besar dari 1 maka daerah A berhasil menularkan virus ke daerah tujuan.



* Jika virus dari daerah awal berhasil tersebar ke daerah tujuan, maka jumlah orang terkena virus penyakit di daerah tujuan akan dirumuskan dengan fungsi I yang sudah dijelaskan sebelumnya.
* Jika dari daerah A berhasil menyebarkan virus ke daerah B, maka akan dicek apakah daerah B mampu menyebarkan virus ke daerah lain yang terhubung dan belum terkena virus penyakit.

Pada awalnya, hanya ada satu daerah saja yang memiliki virus penyakit yang berpotensi menyebar ke daerah lain. Tugas Anda adalah membantu WHO untuk membuatkan simulasi penyebaran virus penyakit, dengan membuat program yang menerima query berupa hari (T) dan menunjukkan daerah mana saja yang sudah tersebar virus penyakit pada hari ke T beserta jalur penyebarannya dengan menggunakan algoritma BFS.

Peta yang menghubungkan daerah-daerah di dunia diberikan dalam berkas eksternal untuk dibaca oleh program dengan format sebagai berikut. Baris pertama pada berkas adalah N yang merupakan banyaknya jalur yang akan diberikan. Sebanyak N baris berikutnya berisi dua buah string (A, B) dan sebuah float (Tr(A,B)). Gambar 1 akan menunjukkan gambar berkas eksternal yang berisi peta yang menghubungkan daerah-daerah di dunia.

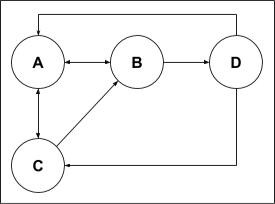


Gambar 1. Berkas eksternal yang berisi peta yang menghubungkan daerah-daerah di dunia.

Populasi masyarakat di suatu daerah di seluruh dunia akan diberikan dalam berkas eksternal untuk dibaca oleh program dengan format sebagai berikut. Baris pertama pada berkas adalah N yang merupakan banyaknya daerah yang terdapat di dunia ini dan sebuah string (A) yang merupakan sumber dari virus penyakit. Sebanyak N baris berikutnya berisi sebuah string (A) dan sebuah integer (P(A)). Gambar 2 akan menunjukkan gambar berkas eksternal yang berisi populasi masyarakat di suatu daerah di seluruh dunia.

Gambar 2. Berkas eksternal yang berisi populasi masyarakat di suatu daerah di seluruh dunia.

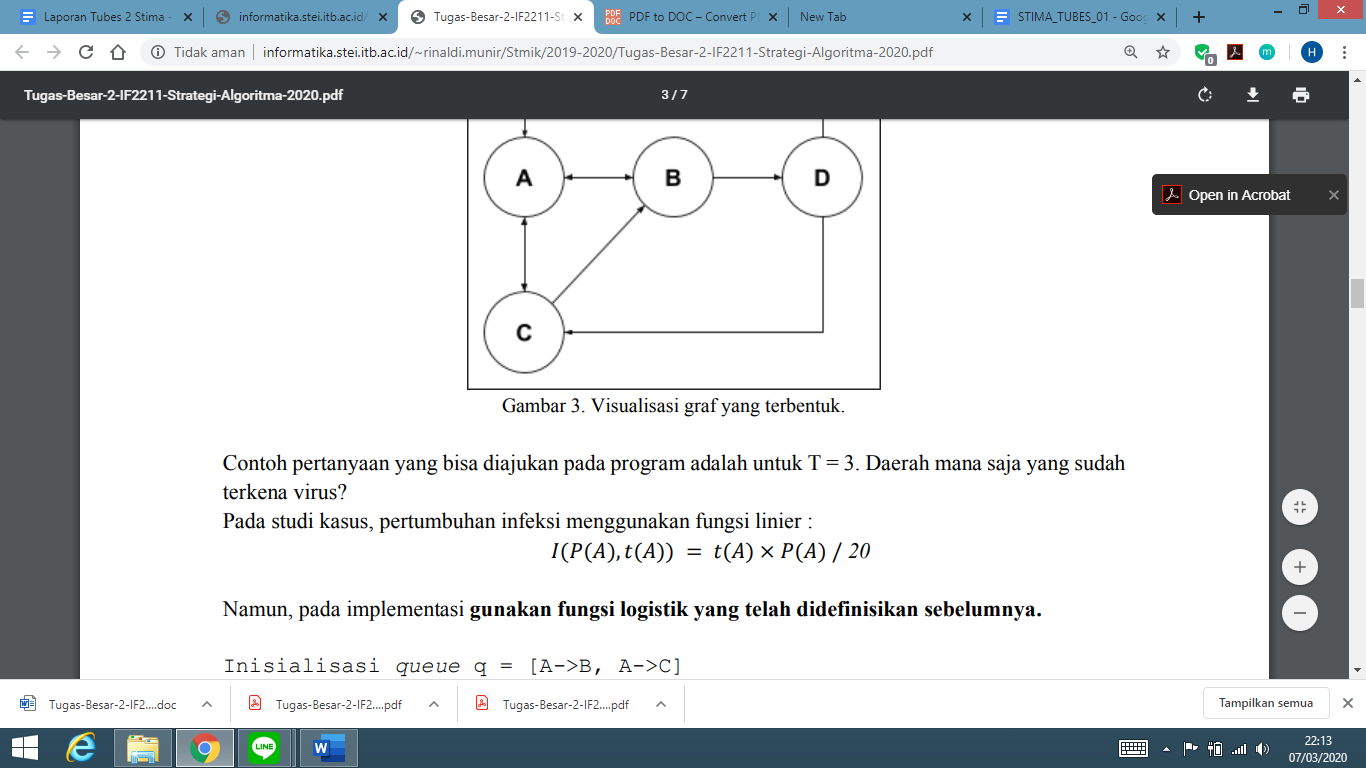
Contoh visualisasi graf yang terbentuk dari berkas eksternal pada gambar 1 dan gambar 2 dapat dilihat pada gambar 3.



Gambar 3. Visualisasi graf yang terbentuk.

Contoh pertanyaan yang bisa diajukan pada program adalah untuk T = 3. Daerah mana saja yang sudah terkena virus?

Pada studi kasus, pertumbuhan infeksi menggunakan fungsi linier :



Namun, pada implementasi **gunakan fungsi logistik yang telah didefinisikan sebelumnya.**

Inisialisasi *queue* q = [A->B, A->C]

❏ **Cek penyebaran dari A -> B**

Virus pertama kali muncul di daerah A pada T(A) = 0 maka t(A) = 3-0 = 3

Jumlah masyarakat yang terkena virus di daerah A adalah I(A, 3) = 3 . 1000 / 20 = 150 S(A,B) = I(A, 3) . Tr(A, B) = 150 . 0.02 = 3

Karena S(A,B) > 1, maka **virus berhasil tersebar dari daerah A ke B.**

Selanjutnya, cari kapan virus berhasil tersebar ke daerah B.

S(A,B) = I(A, t) . Tr(A,B) > 1

= t . 1000 / 20 . 0.02 > 1

= t > 1.

**Maka, virus akan menyebar dari daerah A ke B 2 hari setelah virus menyebar ke A atau T(B) = 2+T(A) = 2+0 = 2.**

q = [A->C, B->A, B->D]

❏ **Cek penyebaran dari A -> C**

Virus pertama kali muncul di daerah A pada T(A) = 0 maka t(A) = 3-0 = 3

Jumlah masyarakat yang terkena virus di daerah A adalah I(A, 3) = 3 . 1000 / 20 = 150 S(A,C) = I(A,3) . Tr(A,C) = 150 . 0.005 = 0.75

Karena S(A,C) <= 1, maka **virus tidak berhasil tersebar dari daerah A ke C**.

q = [B->A, B->D] (Tidak ditambah karena virus tidak berhasil tersebar ke

daerah C)

❏ **Cek penyebaran dari B -> A**

Virus pertama kali muncul di daerah B pada T(B) = 2 maka t(B) = 3-2 = 1

Jumlah masyarakat yang terkena virus di daerah B adalah I(B, 1) = 1 . 5000 / 20 = 250 S(B,A) = I(B,1) . Tr(B,A) = 250 . 0.005 = 1.25

Karena S(B,A), maka **virus berhasil tersebar dari daerah B ke A.**

Selanjutnya, cari kapan virus berhasil tersebar ke daerah A.

S(B,A) = I(B, t) . Tr(B,A) > 1

= t . 5000 / 20 . 0.005 > 1

= t > 0.8.

**Maka, virus akan menyebar dari daerah B ke A 1 hari setelah virus menyebar ke B atau T’(A) = 1+T(B) = 1+ 2 = 3.**

**Namun karena mulanya virus sudah menginfeksi A pada T(A) = 0 (T(A) < T’(A)), maka nilai T(A) tidak berubah.**

q = [B->D] (Tidak ditambah karena T(A) < T’(A))

❏ **Cek penyebaran dari B -> D**

Virus pertama kali muncul di daerah B pada T(B) = 2 maka t(B) = 3-2 = 1

Jumlah masyarakat yang terkena virus di daerah B adalah I(B, 1) = 1 . 5000 / 20 = 250 S(B,D) = I(B,1) . Tr(B,D) = 250 . 0.005 = 1.25

Karena S(B,D), maka **virus berhasil tersebar dari daerah B ke D.**

Selanjutnya, cari kapan virus berhasil tersebar ke daerah D.

S(B,D) = I(B, t) . Tr(B,D) > 1

= t . 5000 / 20 . 0.005 > 1

= t > 0.8.

**Maka, virus akan menyebar dari daerah B ke D 1 hari setelah virus menyebar ke D atau T(D) = 1+T(B) = 1+ 2 = 3.**

q = [D->A, D->C]

❏ **Cek penyebaran dari D -> A**

Virus pertama kali muncul di daerah D pada T(D) = 3 maka t(D) = 3-3 = 0

Jumlah masyarakat yang terkena virus di daerah D tentu saja 0 \* sehingga S(D,A)) = 0

Karena S(D,A) <= 1, maka **virus tidak berhasil tersebar dari daerah D ke A.**

q = [D->C]

❏ **Cek penyebaran dari D -> C**

Virus pertama kali muncul di daerah D pada T(D) = 3 maka t(D) = 3-3 = 0

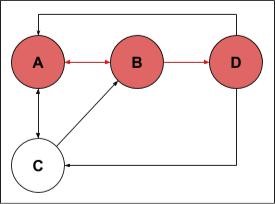
Jumlah masyarakat yang terkena virus di daerah D tentu saja 0 \* sehingga S(D,C) = 0

Karena S(D,C) <= 1, maka **virus tidak berhasil tersebar dari daerah D ke C.**

q = []

\*) Walaupun jumlah masyarakat yang terkena virus di daerah D adalah 0, namun daerah D dianggap sudah terinfeksi karena virus telah berhasil tersebar dari daerah B.

Maka daerah yang sudah terkena virus penyakit adalah A, B, dan D dengan jalur infeksi antara lain sebagai berikut A->B, B->A, B->D dengan visualisasi yang dapat dilihat pada gambar 4 berikut.



Gambar 4. Visualisasi hasil graf yang terbentuk.

End of program

Berikut adalah *pseudocode* dari program BFS untuk melakukan simulasi penyebaran virus penyakit



Program yang dibuat harus memenuhi spesifikasi sebagai berikut.

1. Aplikasi bisa menerima peta daerah-daerah di seluruh dunia berikut populasinya seperti contoh pada Gambar 1 dan Gambar 2.
2. Dari berkas eksternal yang telah diterima, aplikasi dapat menampilkan visualisasi graf peta daerah-daerah di seluruh dunia. Proses visualisasi ini boleh memanfaatkan pustaka atau kakas yang tersedia. Sebagai referensi , salah satu kakas yang tersedia untuk melakukan visualisasi adalah MSAGL **:** <https://github.com/microsoft/automatic-graph-layout>
3. Aplikasi bisa menerima pertanyaan (query) berupa jumlah hari. Pertanyaan akan diterima melalui aplikasi berbasis GUI yang dibangun sesuai kreativitas masing-masing.
4. Dari graph yang sudah dibentuk, aplikasi harus dapat menyusun jalur infeksi. Aplikasi juga harus dapat menunjukkan langkah-langkah proses penentuan daerah-daerah yang terinfeksi dengan memanfaatkan algoritma BFS. Mahasiswa **tidak diperkenankan** untuk melihat atau menyalin library lain yang mungkin tersedia bebas terkait dengan pemanfaatan BFS.
5. Aplikasi harus dapat membedakan simpul untuk kota yang sudah terinfeksi dan yang belum terinfeksi berikut juga dengan jalurnya.
6. Data Uji akan diberikan oleh asisten.

# BAB II

# DASAR TEORI

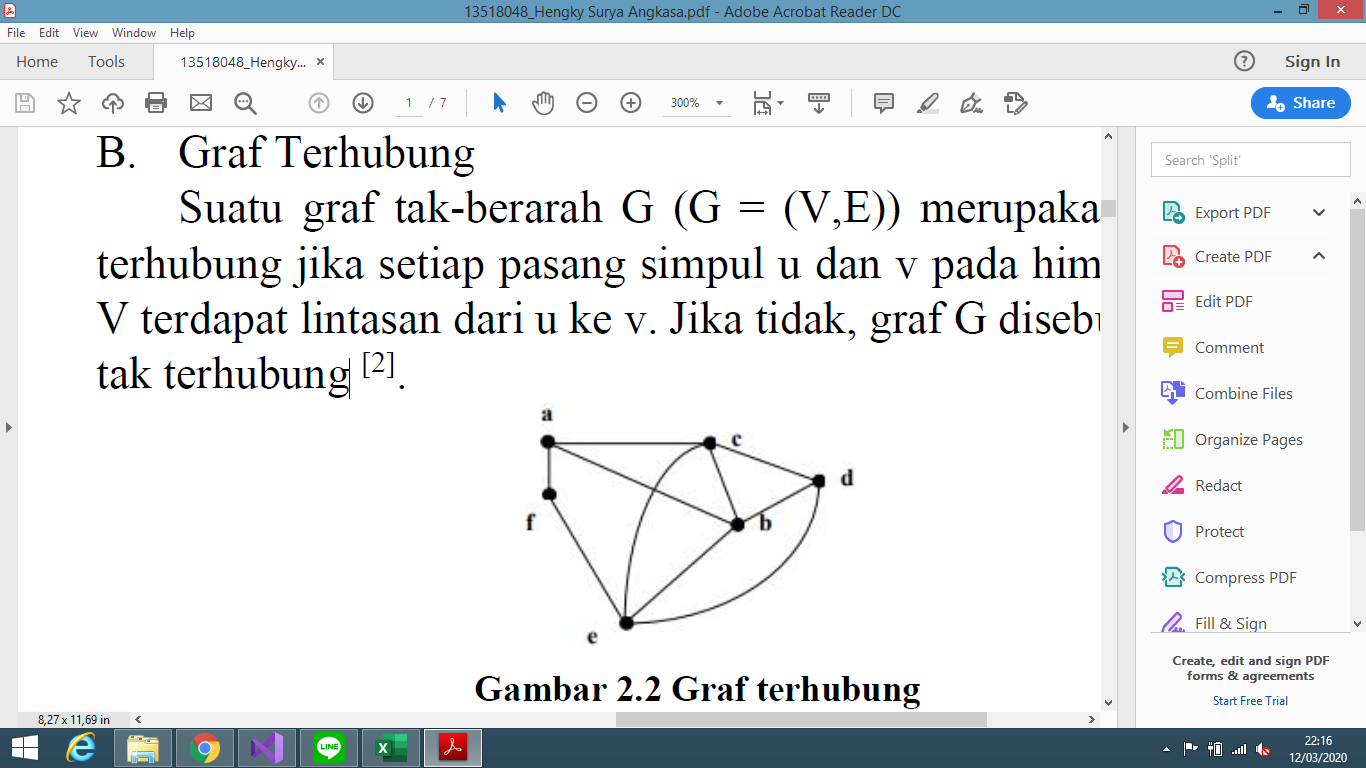
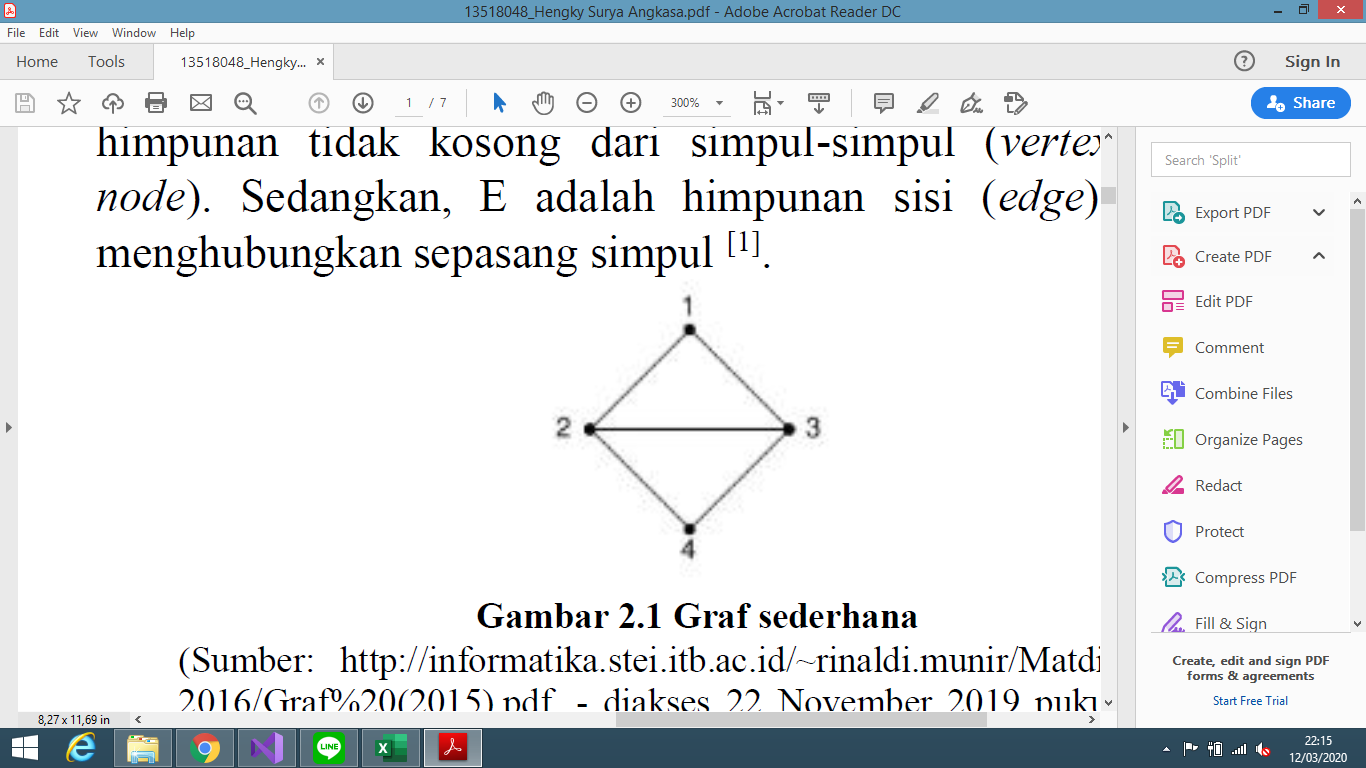
## Algoritma Breadth First Search

Algoritma ini merupakan algoritma pencarian melebar. Misalnya simpul *v* merupakan simpul awal yang hendak dikunjungi. Pertama, kunjungi simpul *v*. Kedua, kunjungi semua simpul yang bertetangga dengan simpul *v,* masukkan simpul yang dikunjungi ke antrian*.* Terakhir, kunjungi tetangga tiap simpul yang ada di antrian.

BFS memakai beberapa struktur data. Matriks ketetanggan, misalnya A. Elemen Aij bernilai 1 jika simpul i dan j bertetangga dan bernilai 0 jika simpul i dan j tidak bertetangga. Antrian atau queue (q) untuk menyimpan simpul tetangga yang belum dikunjungi. Tabel atau *array of boolean* (T) untuk menyimpan apakah suatu simpul i telah dikunjungi, T[i] bernilai *true*, atau simpul i belum dikunjungi, T[i] bernilai *false*.

## Graf

Graf G merupakan pasangan himpunan (V,E). V adalah himpunan tidak kosong dari simpul-simpul (*vertex* atau *node*). Sedangkan, E adalah himpunan sisi (edge) yang menghubungkan sepasang simpul. Suatu graf tak-berarah G (G = (V,E)) merupakan graf terhubung jika setiap pasang simpul u dan v pada himpunan V terdapat lintasan dari u ke v. Jika tidak, graf G disebut graf tak terhubung.



**Gambar 2.1 Graf sederhana**

# BAB III

# ANALISIS PEMECAHAN MASALAH

## Struktur Data dan Spesifikasi Program

Program ditulis dalam bahasa C#, dikembangkan dengan IDE Visual Studio, dan visualisasi menggunakan *library* msagl. Adapun program memakai kakas .NET. Struktur data utama yang digunakan untuk menyelesaikan masalah pada Tugas Besar ini adalah struktur data Graph. Terdapat beberapa data yang disimpan dalam kelas Graph yang dibuat, yaitu:

1. **Dictionary<string, List<Tuple<string, double>>> graph**

Bagian ini adalah representasi graf berupa adjacency list, dengan *key* adalah *source* dan *value* adalah list dari *tuple* (destination, Tr) (Tr adalah peluang orang melakukan perjalanan dari source ke destination).

1. **Dictionary<string, int> population**

Bagian ini dipakai untuk menyimpan suatu simpul (daerah) dan jumlah populasi dari daerah tersebut.

1. **Dictionary<string, int> timeInfected**

Bagian ini dipakai untuk menyimpan waktu pertama kali suatu daerah terinfeksi oleh virus.

1. **Dictionary<string, List<string>> edgeInfected**

Bagian ini dipakai untuk menyimpan daftar edge yang menjadi jalan penyebaran virus.

1. Selain itu, juga disimpan data inputTime dan viralSource.

## Langkah-Langkah Pemecahan Masalah

Masalah yang diberikan pada Tugas Besar ini adalah mengenai simulasi penyebaran virus memanfaatkan algoritma Breadth First Search yang sedikit dimodifikasi. Secara umum, langkah-langkah pemecahan masalah tersebut sudah diberikan pada spesifikasi tugas dalam bentuk *pseudocode* (dapat dilihat pada Bab I laporan ini). Sebagian besar *pseudocode* tersebut diimplementasikan dalam prosedur BFS yang terletak di dalam kelas Graph. Langkah-langkah pemecahan masalahnya adalah sebagai berikut (langkah 2-9 dilakukan dalam prosedur BFS):

1. Baca input melalui *Graphical User Interface* yang telah dibuat. Input yang diberikan berupa input dua buah file dan input waktu (*t*).
2. Format isi kedua file masukan dijelaskan pada spesifikasi tugas (lihat Bab I). Proses pembacaan file masukan dilakukan menggunakan metode static readGraphFromFile yang berada di dalam kelas FileReader.
3. Input waktu akan diset ke dalam graf yang telah terbentuk dari proses sebelumnya.

Langkah ini akan menghasilkan suatu graf yang siap dipakai.

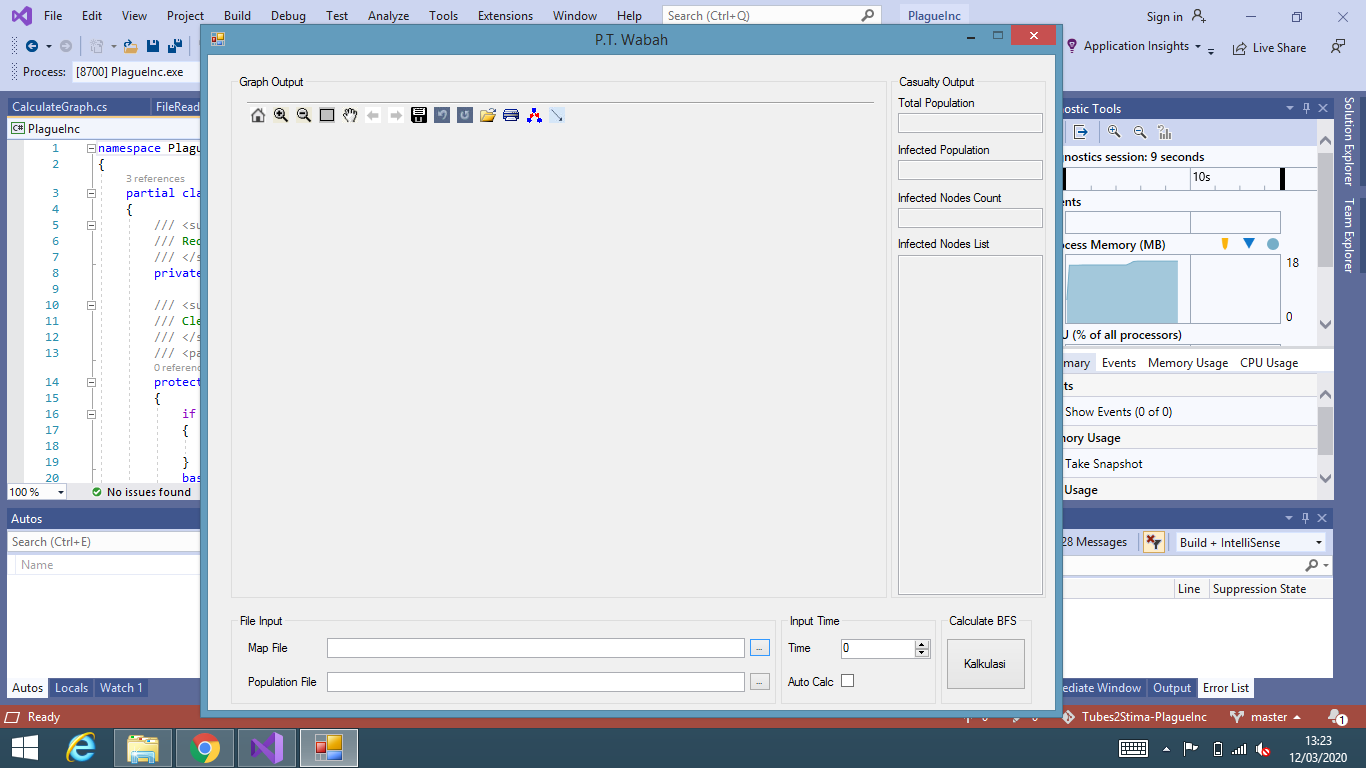
1. Inisialisasi timeInfected dengan int.MaxValue sebagai pengganti nilai infiniti.
2. Misalkan *v* adalah simpul yang berperan sebagai sumber wabah (viralSource), set timeInfected[v] = 0.
3. Masukkan semua tetangga *v* ke antrian (*q*) dalam bentuk tuple (*v*, *tetangga\_v*).
4. Selama antrian *q* belum kosong, ulangi langkah 6-9.
5. Keluarkan head dari antrian *q*, sehingga didapat tuple(asal, tujuan).
6. Hitung apakah fungsi penyebaran S(asal, tujuan) bernilai lebih dari 1.
7. Proses ini bergantung proses sebelumnya.
8. Jika langkah 7 bernilai benar, cari waktu (*t*) pertama kali virus tersebar ke kota tujuan. Proses ini dilakukan dengan melakukan pengecekan nilai *t* dari 0 hingga *tmax* sampai didapat nilai S(asal, tujuan, t) > 1.   
   Sebagai catatan, *tmax* diimplementasikan dalam fungsi t(string A) yang mengembalikan nilai Math.Max(0, inputTime - timeInfected[A]). Selain itu, perhatikan bahwa kami melakukan overloading metode S dengan kesamaan S(asal, tujuan) = S(asal, tujuan, tmax).
9. Jika langkah 7 salah, lanjut ke iterasi berikutnya (langkah 6-9).
10. Jika timeInfected[asal] + t <= timeInfected[tujuan], lakukan proses berikut:
11. Set timeInfected[tujuan] = timeInfected[asal] + t.
12. Catat edge (asal, tujuan) ke variabel edgeInfected.
13. Masukkan semua simpul tetangga simpul tujuan ke antrian.
14. Konversi graf yang telah diproses menjadi graf msagl. Proses ini dilakukan dengan metode graphConverter di dalam kelas GraphConverter.
15. Tampilkan graf hasil ke GUI.

# BAB IV

# IMPLEMENTASI DAN PENGUJIAN

## Implementasi Program

Dalam program yang kami buat, terdapat beberapa *file* sebagai implementasi dari persoalan. Pertama, *FileReader.cs* yang digunakan untuk membaca file eksternal (.txt), yaitu file peta yang menghubungkan daerah-daerah di dunia dan *file* populasi masyarakat di suatu daerah di dunia. Kedua, *Frontend.cs* yang berisi program untuk menampilkan visualisasi graf ke layar dan menerima *path file* peta dan populasi, menerima input waktu (T) dan menampilkan beberapa informasi.

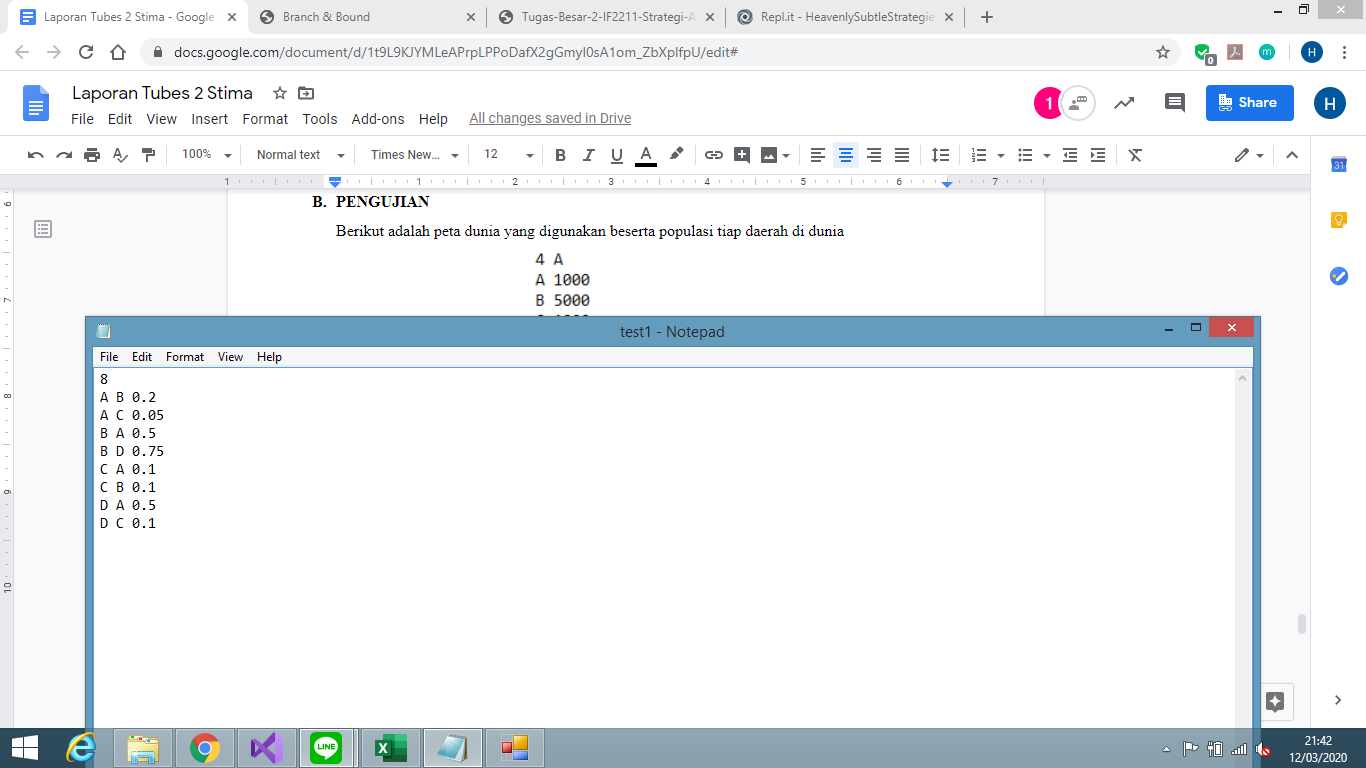
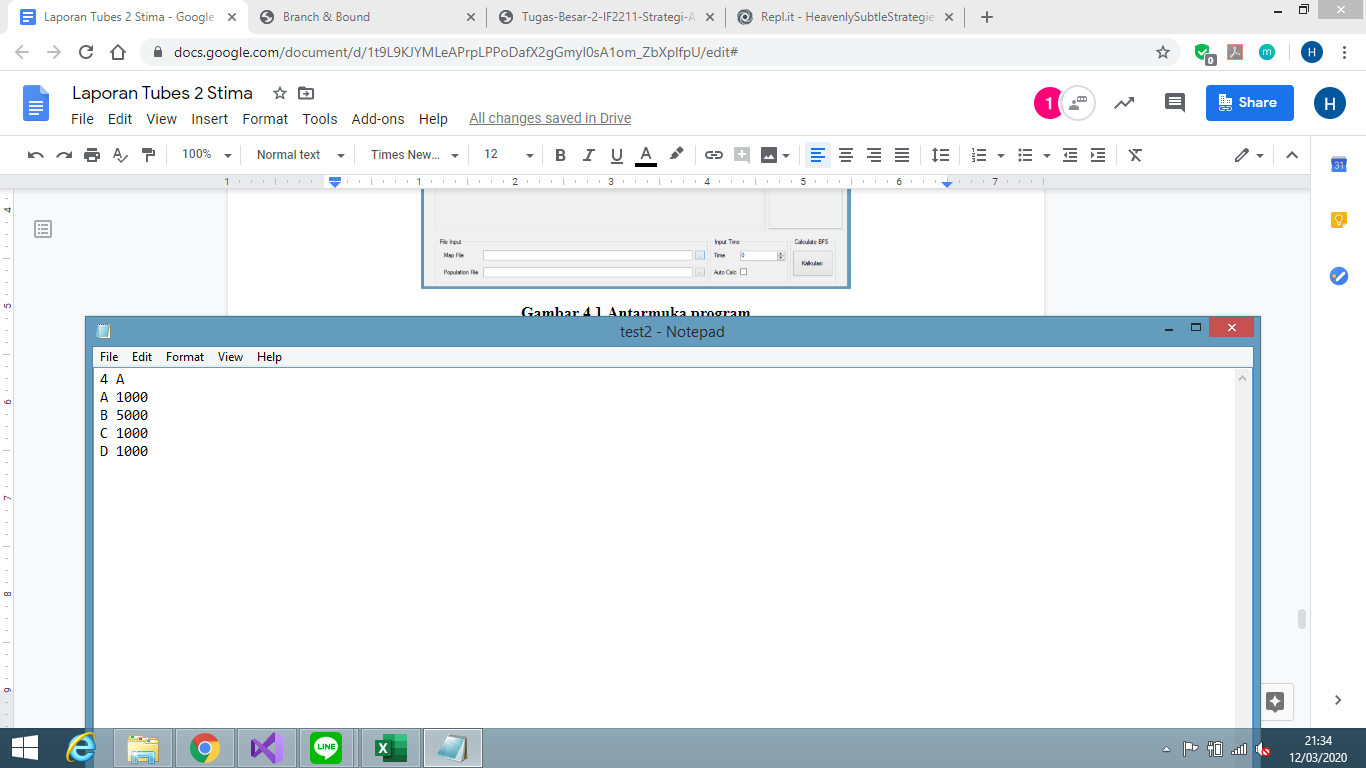
****

**Gambar 4.1 Antarmuka program**

Ketiga, *Graph.cs* berisi Kelas Graph yang dipakai dalam algoritma BFS. Keempat, *GraphConverter.cs* digunakan untuk melakukan konversi graf yang dihasilkan dari algoritma BFS menjadi graf *msagl*.

## Pengujian

Pengujian dilakukan dengan menerima input *path* dua *file external* yang berisi peta dan populasi daerah. Secara *default*, *input time* bernilai 0. *Input time* dapat diganti sesuai dengan keinginan. Lalu, tekan kalkulasi untuk menampilkan graf penyebaran virus sesuai *input* *time*. Daerah (simpul) dan jalur (sisi) akan berwarna merah jika suatu daerah telah terinfeksi virus dan jalur antar daerah (sisi) merupakan jalur penyebaran virus.

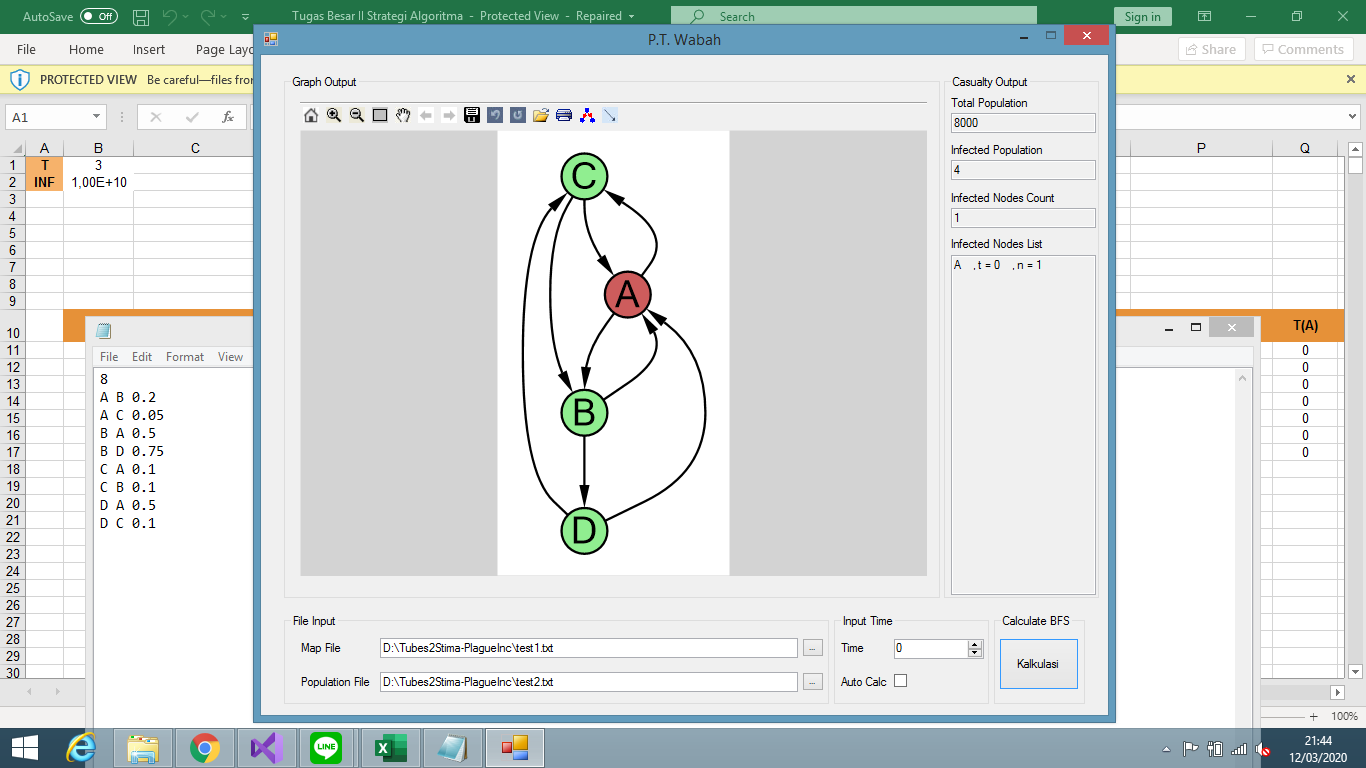


**Gambar 4.2 Peta dunia dan populasi**

Berikut adalah hasil saat t = 0. Dapat dilihat pada panel kanan bagian *Infected Population* sudah terdapat 4 orang yang terhitung *infected*, lebih tepatnya 1 orang di tiap node. Hal ini dapat dijelaskan karena fungsi I(A,t(A)) yang digunakan adalah

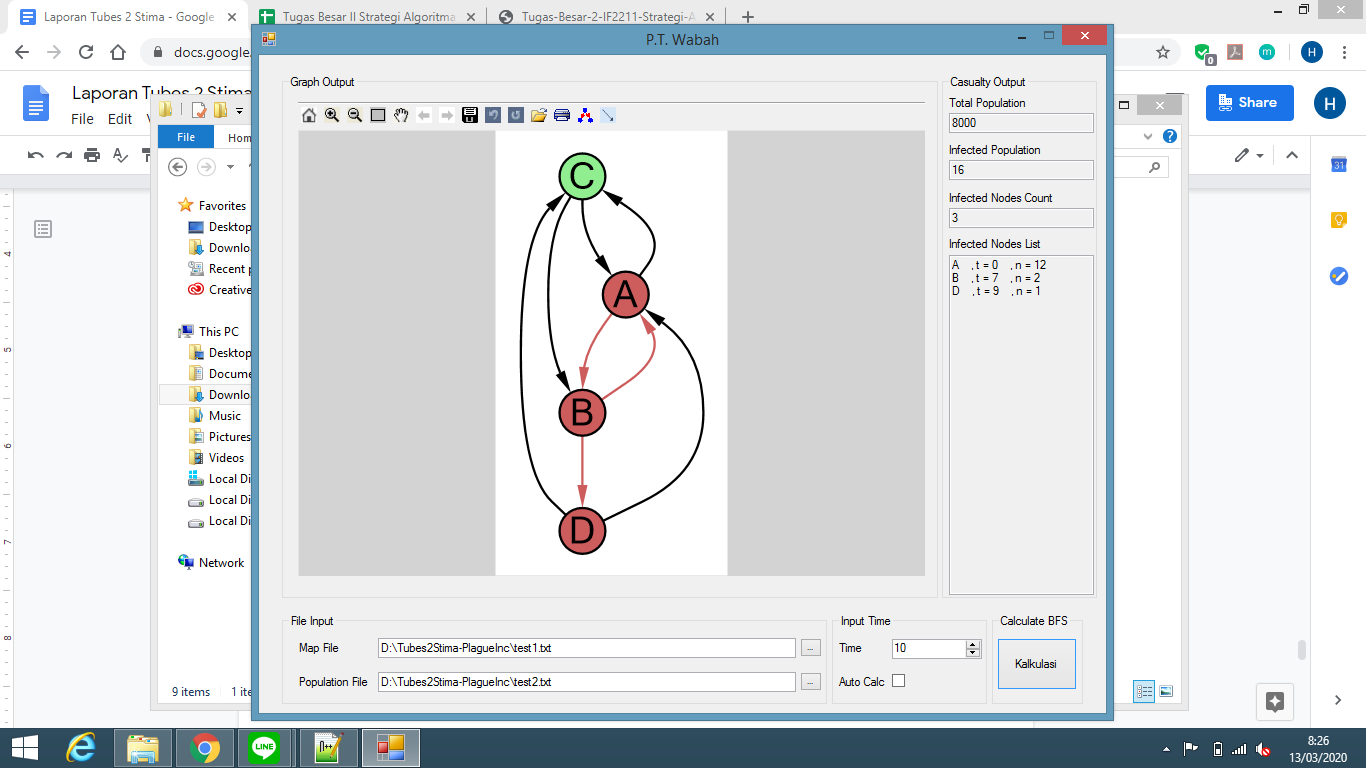
****

Sehingga saat t(A) = 0, nilai I(A) akan bernilai 1 di node manapun (dengan P(A) != 0).



**Gambar 4.3 Graf BFS saat t = 0**

Berikut adalah hasil saat t = 10. Keluaran yang dihasilkan sudah sesuai dengan keluaran yang diberikan di *sheet debug* yang disediakan asisten (<https://docs.google.com/spreadsheets/d/1lXuxTMSS08Isijf8ETc0EIvW_srnxFQg0hRHEL_e5fY/edit#gid=0>). Keluaran tersebut berupa *node* yang terinfeksi, *edge* penyebaran, waktu pertama kali infeksi, dan jumlah infeksi tiap node. Sedangkan untuk jumlah infeksi total seperti dijelaskan sebelumnya, kota yang memiliki t(A) = 0, akan dianggap memiliki 1 infected. Sebagai catatan, *edge* penyebaran dicatat juga *edge* yang mengarah ke kota yang sudah terinfeksi dengan S(src,dst) > 1, namun *timeInfected* pada kota yang sudah terinfeksi tersebut tidak berubah. *TimeInfected* hanya akan berubah ketika timeInfected[src] + time <= timeInfected[dst].

****

**Gambar 4.5 Graf BFS saat t = 10**

# 

# BAB V

# KESIMPULAN DAN SARAN

## Kesimpulan

Simpulan yang dapat kami ambil dalam pembuatan tugas besar ini adalah:

1. Algoritma Breadth First Search (BFS) dapat diimplementasikan dalam simulasi penyebaran virus pada penelusuran graf dengan modifikasi pada algoritma BFS yang umum.
2. Algoritma BFS memberikan solusi dengan langkah minimum. Namun demikian, algoritma BFS memiliki kompleksitas ruang yang kurang baik daripada algoritma DFS.

## Saran

Dari proses pembuatan tugas besar ini, terdapat saran yang dapat diterapkan.

1. Penggunaan kakas yang memiliki *support* lebih terhadap pengembangan aplikasi di beberapa platform, mengingat .NET Framework lebih ditujukan untuk mengembangkan aplikasi Windows.
2. Proses dan persamaan yang digunakan dalam simulasi penyebaran dibuat lebih detail, sehingga hasil yang diperoleh pun akan menjadi semakin akurat.
3. Menggunakan fungsi I(A, t(A)) yang lain karena yang digunakan pada tugas besar ini nilai fungsi tersebut minimal satu, sehingga pada t = 0 yang terinfeksi 1 per simpul yang ada padahal yang terinfeksi baru 1 simpul (simpul awal penyebaran).

# 

# DAFTAR PUSTAKA

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-(2020).pdf>

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Tugas-Besar-2-IF2211-Strategi-Algoritma-2020.pdf>