SURELY YOU CONGEST

(SIMULASI MANAJEMEN LALU LINTAS UNTUK MENGGURANGI KEMACETAN DI PERSIMPANAGAN DAERAH KABUPATEN SUMEDANG)

Kelompok 1 Kelas IF-39-03:
Fedy Fahron Guntara (1301160192)
Aan Saputra (1301150045)
T hablumminallah (1301154333

ABSTRAK

Tingginya volume lalulintas yang melintasi jalan raya di kabupaten Sumedang khususnya mobil cenderung menimbulkan kemacetan pada beberapa persimpangan jalan terutama pada saat jam puncak pagi hari. Manajemen lalulintas merupakan salah satu cara untuk mengatasi masalah tersebut. Penulisan ini bertujuan untuk mengetahui seberapa besar pengaruh penerapan manajemen lalulintas untuk mengurangi kemacetan di jalan raya di kabupaten Sumedang yang dilalui oleh mobil. Penulisan ini menggunakan metode *queue* dan *graf* yang ada pada struktur data. Selanjutnya dilakukan simulasi manajemen lalulintas di jalan raya yang di lalui mobil dengan mempergunakan perangkat lunak C++ yang sudah dicoding program simulasi. Berdasarkan hasil simulasi diperoleh solusi alternatif manajemen lalulintas melalui pemanfaatan ruas jalan lain yang sudah ada sebagai alternatif rute mobil sehingga diperoleh berkurangnya antrian dan penumpukan serta waktu temput.

Kata kunci: *Manajemen lalulintas, Trafikplan.*

BAB I
PENDAHULUAN

Tingginya volume lalulintas yang melintasi jalan raya di kabupaten Sumedang khususnya mobil cenderung menimbulkan kemacetan pada beberapa persimpangan jalan terutama pada saat jam puncak pagi hari. Manajemen lalulintas merupakan salah satu cara untuk mengatasi masalah tersebut. Penulisan ini bertujuan untuk mengetahui seberapa besar pengaruh penerapan manajemen lalulintas untuk mengurangi kemacetan di jalan raya di kabupaten Sumedang yang dilalui oleh mobil . Tujuannya adalah membandingkan berbagai solusi alternatif penerapan manajemen lalulintas untuk mengurangi penumpukan serta antrian di jalur rute (mobil) di kabupaten Sumedang.

Penulisan ini menggunakan metode *queue* dan *graf* yang ada pada struktur data. Queue pada Struktur Data atau antrian adalah sekumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung disebut dengan sisibelakang(rear), dan penghapusan(pengambilan elemen) dilakukan lewat ujung lain (disebut dengan sisi depan atau front).

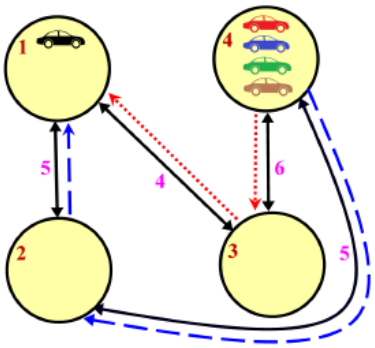
Eksperimen ini dilakukan satu kali, tetapi dengan berbagai inputan dapat di simulasikan. Dengan kata lain eksperimen ini dilakukan setiap ada inputan baru yang ingin di simulasikan.

BAB II
DESKRIPSI MASALAH DAN ASUMSI DALAM PEMODELAN

Permasalahan ini mengenai cara untuk merancang sistem manajemen lalu lintas terpusat untuk mobil pintar. Itu tujuannya adalah untuk menggunakan informasi global untuk menginstruksikan para komuter pagi, yang harus berkendara ke pusat kota dari pinggiran kota, cara terbaik untuk sampai ke pusat kota sambil menghindari kemacetan. simulasi hanya bisa meyakinkan mereka untuk mengubah ke rute yang berbeda yang sama cepatnya.

Jaringan jalan kota terdiri dari persimpangan yang dihubungkan oleh jalan dua arah dari berbagai waktu perjalanan. Setiap komuter mulai di beberapa persimpangan, yang dapat bervariasi dari komuter ke komuter. Semua penumpang mengakhiri perjalanan mereka di tempat yang sama, yaitu pusat kota di persimpangan 1. Jika dua komuter mencoba untuk memulai perjalanan di sepanjang jalan yang sama ke arah yang sama pada saat yang sama waktu, akan ada kemacetan; Anda harus menghindari ini. Namun demikian, itu baik jika dua penumpang lewat persimpangan yang sama secara bersamaan atau jika mereka mengambil jalan yang sama mulai pada waktu yang berbeda.

Simulasi ini bertujuan untuk menentukan jumlah maksimum komuter yang dapat berkendara ke pusat kota tanpa kemacetan, subjek untuk semua komuter yang memulai perjalanan mereka pada waktu yang sama dan tanpa mereka mengambil rute sub optimal.



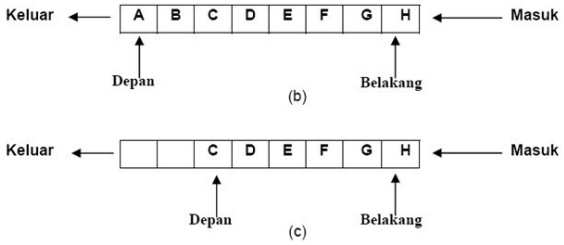
Gambar 1. Ilustrasi

Dalam simulasi ini ada beberpa asusmsi yang dapat melancarkan simulasi ini. Asumsi ini berhubungan dengan rute yang akan ditempuh yaitu :

1. Diasumsikan bahwa setiap rute yang dilalui memiliki volume yang sama
2. Diasumsikan bahwa tidak ada keegoisan pengendara
3. Diasumsikan tidak ada kemacetan yang disebabkan oleh lain hal (contoh : kecelakaan dll.).
4. Kecepatan setiap kendaraan sama.

BAB III
DASAR TEORI DAN STUDI LITERATUR

Queue pada Struktur Data atau antrian adalah sekumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung disebut dengan sisi belakang(rear), Pada Queue atau antrean Terdapat satu buah pintu masuk di suatu ujung dan satu buah pintu keluar di ujung satunya di mana membutuhkan variabel Head dan Tail (depan/front, belakang/rear).



Gambar 2. queue

Karakteristik Queue atau antrean :

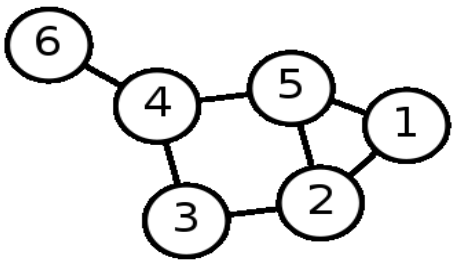
1. Elemen antrean
2. Front (elemen terdepan antrean)
3. Tail (elemen terakhir)
4. Jumlah elemen pada antrean
5. Status antrean

Operasi pada Queue atau antrean

1. Tambah(menambah item pada belakang antrean)
2. Hapus (menghapus elemen depan dari antrean)
3. Kosong(mendeteksi apakah pada antrean mengandung elemen atau tidak)

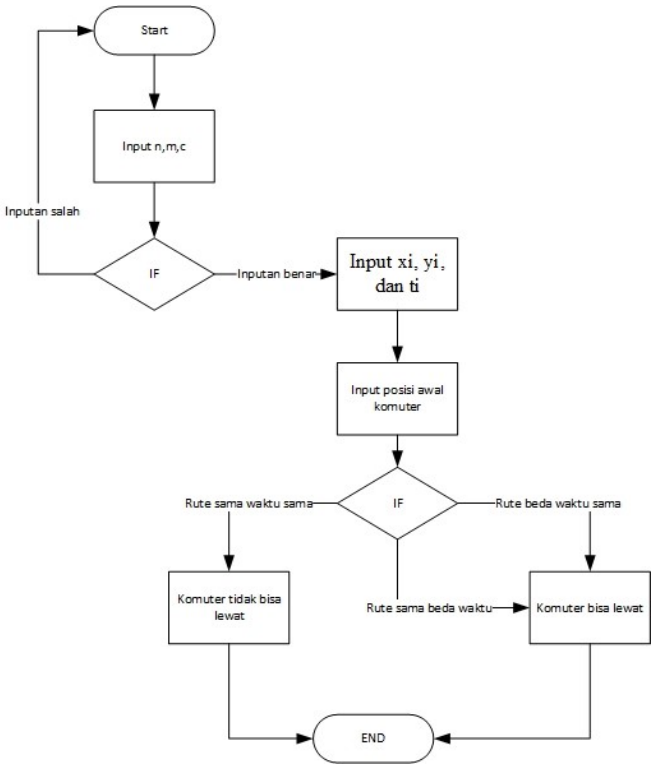
Graf adalah himpunan dari objek-objek yang dinamakan titik, simpul, atau sudut dihubungkan oleh penghubung yang dinamakan garis atau sisi. Sebuah **graf** atau **graf tidak berarah** G adalah sebuah pasangan G:=(V,E) yang memenuhi kondisi:

1. V adalah sebuah himpunan , yang elemennya dinamakan **sudut** atau **simpul**
2. E adalah sebuah himpunan dari pasangan-pasangan sudut yang terpisah, yang dinamakan **sisi** atau **garis**.



Gambar 3. graf

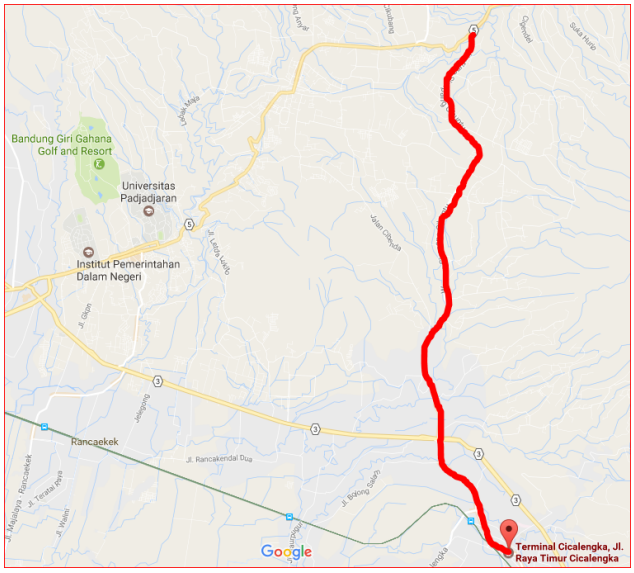
Masukan terdiri dari beberapa uji kasus. Baris pertama berisi tiga bilangan bulat n, m, dan c, di mana n (1 <n <25000) adalah jumlah persimpangan, m (0 <m <50000) adalah jumlah jalan dan c (0 <c <1000) adalah jumlah komuter. Setiap baris m berikutnya berisi tiga bilangan bulat xi, yi, dan ti menggambarkan satu jalan, di mana xi dan yi (1 <xi; yi <n) adalah persimpangan yang berbeda yang menghubungkan jalan, dan ti (1 <ti <10000) adalah waktu yang diperlukan untuk melakukan perjalanan di sepanjang jalan itu di kedua arah. Kamu boleh asumsikan bahwa pusat kota dapat dijangkau dari setiap persimpangan. Baris terakhir berisi c integer yang mencantumkan mulai persimpangan dari komuter.



Gambar 3. Flowmap

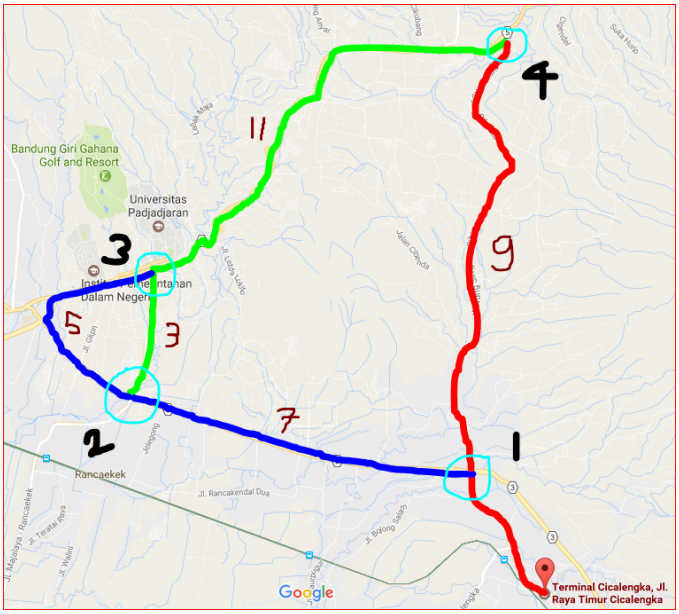
BAB IV
PEMODELAN SISTEM DAN ANALISISNYA

Selain menggunakan data yang ada, penulis juga membuat data untuk menguji apakah program dapat berjalan sesuai dengan yang diharapkan. Penulis menggunakan jalur mobil yang ada di daerah kabupaten Sumedang sebagai data yang akan di *test*.



Gambar 4. Jalur di Kabupaten Sumedang

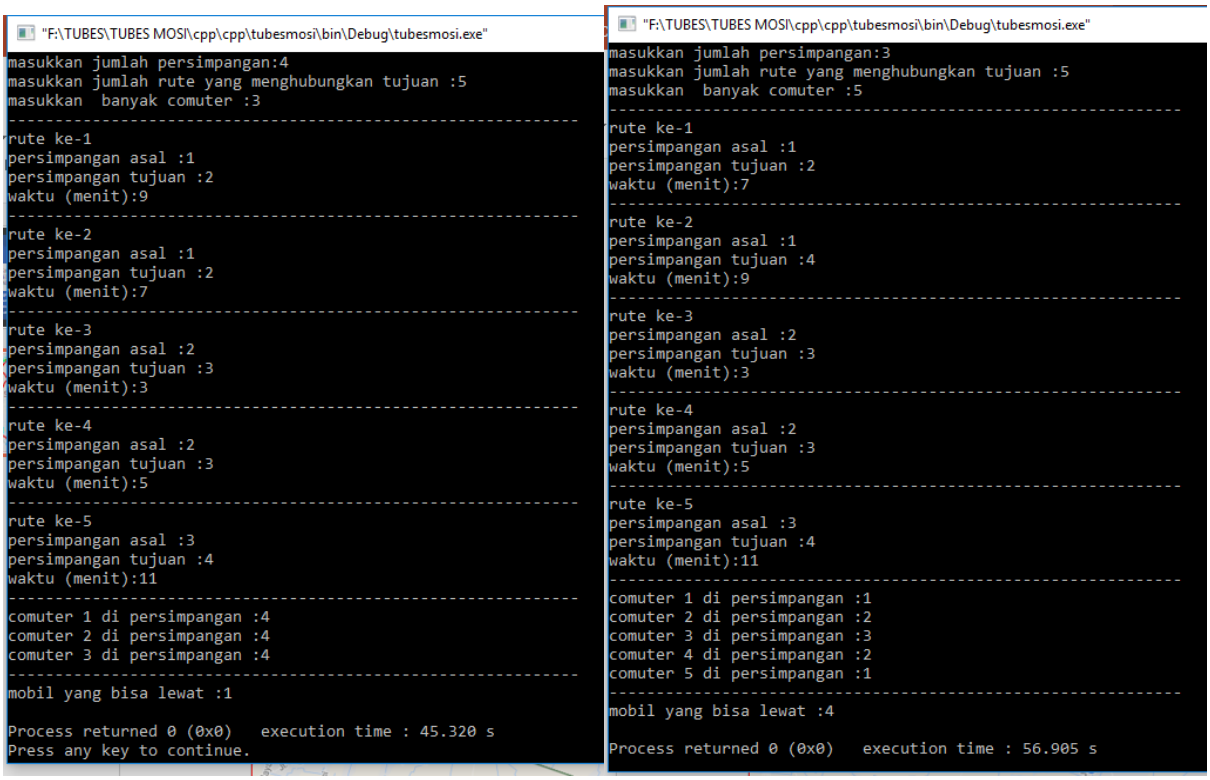
Gambar 4 merupakan jalur mobil yang selama ini beroperasi. Garis merah menunjukan rute yang di ambil oleh mobil. Untuk tujuan pengujian maka penulis menambahkan beberapa perubahan sehingga bisa dilihat perbedaannya.



Gambar 5. Jalur yang sudah dimofikasi untuk melakukan *test*

Gambar 5 merupakan gambar rute mobil yang sudah siap dijadikan test dengan inputan :

Inputan 1	Inputan 2
4 5 3 1 4 9 1 2 7 2 3 3 2 3 5 3 4 11 4 4 4	3 5 5 1 2 7 1 4 9 2 3 3 2 3 5 3 4 11 1 2 3 2 1



Gambar 6. Contoh inputan dan outputan

BAB V
KESIMPULAN

Program ini telah dibuat dengan menggunakan metode *Queue* dan *Graf* yang sudah dijabarkan diatas yang dimaksudkan untuk mengatur jalanya lalulintas dan mengantisipasi kemacetan yang sering terjadi yang disebabkan oleh padatnya kendaraan. Kita menggunakan metode tersebut karena *Queue* dan *Graf* adalah metode yang paling tepat untuk mengatasi masalah *Surely You Congest.*

Referensi

[1] http://darkzone7.blogspot.co.id/2013/04/struktur-ogranisasi-data-2-queue-antrean.html Diakses tanggal 03 Mei 2018 pukul 18.31 WIB

[2] https://id.wikipedia.org/wiki/Graf_(matematika) Diakses tanggal 03 Mei 2018 pukul 21.31 WIB

Lampiran

```
int main() {
 int n, m, c;
 printf("masukkan jumlah persimpangan:");
 scanf("%d",&n);
 printf("masukkan jumlah rute yang
menghubungkan tujuan :");
 scanf("%d",&m);
 printf("masukkan  banyak comuter :");
 scanf("%d",&c);
printf("----------------------------------------------------\n");
 while ((n && m && c) >= 1) {

  for (int i = 1; i <= n; i++)
   g[i].clear();
  for (int i = 0; i < m; i++) {
```

```
      printf("rute ke-%i\n",i+1);
      printf("persimpangan asal :");
  scanf("%d",&X[i]);
  printf("persimpangan tujuan :");
  scanf("%d",&Y[i]);
  printf("waktu (menit):");
  scanf("%d",&T[i]);
  printf("----------------------------------------------------\n");
   //scanf("%d %d %d", &X[i], &Y[i], &T[i]);
   g[X[i]].push_back(make_pair(Y[i], T[i]));
   g[Y[i]].push_back(make_pair(X[i], T[i]));
  }
  for (int i = 0; i < c; i++){
printf("comuter %i di persimpangan :",i+1);
   scanf("%d", &C[i]);
  }
```