

Penerapan Algoritma A* (A star) pada Aplikasi Waze

Aya Aurora Rimbamorani 13515098

Program Studi Informatika

Sekolah Teknik Elektro dan Informatika

Lnstitut Teknologi Bandung, Jl Ganesha 10 Bandung 40132, Indonesia

13515098@std.stei.itb.ac.id

Abstrak—Mobilisasi merupakan salah satu kebutuhan manusia yang sangat vital. Seseorang dalam sehari dapat melakukan perpindahan dari satu tempat ke tempat lain dengan sangat sering. Perpindahan tersebut dapat dilakukan dengan hanya berjalan kaki atau berkendara. Ketika melakukan perpindahan tersebut, seseorang akan melakukannya dengan mudah apabila ia telah mengetahui jalan dari tempat awal ke tempat tujuan. Namun, perpindahan akan menjadi sulit apabila seseorang tidak mengetahui lokasi tujuan dari tempat awal ia mulai berpindah. Dengan adanya aplikasi *waze*, perpindahan dengan lokasi tujuan yang tidak diketahui ini dapat diatasi. Aplikasi *waze* akan menunjukkan jalur terpendek dengan waktu tempuh terkecil dari lokasi awal menuju lokasi tujuan. Penentuan jalur terpendek dengan waktu tempuh terkecil tersebut akan ditinjau dengan menggunakan algoritma A* yang memiliki fungsi heuristik tertentu.

Kata kunci—Aplikasi Waze, fungsi heuristik, algoritma A*

I. LATAR BELAKANG

Mobilitas yang tinggi merupakan salah satu tuntutan bagi manusia abad ke-21. Mobilitas tinggi tersebut dibutuhkan di berbagai aktivitas manusia seperti pendistribusian suatu barang, *meeting* dengan klien, liburan, dan berbagai aktivitas lainnya. Dengan kebutuhan mobilitas yang tinggi tersebut,, manusia diharapkan dapat melakukan perpindahan dari satu tempat ke tempat lainnya dengan aman dan cepat sehingga manusia dapat menyelesaikan suatu aktivitas dan melanjutkan aktivitas lainnya. Perpindahan tersebut akan mudah dilakukan apabila jalur menuju lokasi yang dituju diketahui. Namun, adakalanya suatu tempat yang akan dituju tersebut merupakan tempat baru yang belum pernah dikunjungi sehingga hal tersebut dapat menghambat mobilitas manusia.

Aplikasi *Waze* merupakan suatu aplikasi yang berbasis android, iOS, BlackBerry, Symbian, dan Windows Mobile yang mampu mempermudah mobilitas manusia dengan cara menunjukkan jalan dengan waktu tempuh terkecil dari suatu tempat ke tempat tujuan. Aplikasi ini memanfaatkan sistem navigasi menggunakan GPS (*Global Positioning System*) untuk menentukan lokasi seseorang dan lokasi tujuan. Setelah penentuan lokasi awal dan lokasi tujuan tersebut, barulah aplikasi *Waze* ini melakukan kalkulasi jalur tempuh terpendek dengan waktu tempuh terkecil.

Kalkulasi jalur tersebut dapat dilakukan dengan menerapkan berbagai algoritma. Pada pembahasan kali ini, penulis akan meninjau kalkulasi jalur tempuh terpendek

dengan waktu tempuh terkecil menggunakan algoritma A*. Penerapan algoritma A* tidak hanya meninjau suatu tempat dari jarak tempat awal ke tempat tujuan, namun juga menggunakan suatu fungsi heuristik yang dapat mengoptimalkan jalur terpendek dengan waktu tempuh terkecil.

II. LANDASAN TEORI

A. Graf¹

1) Teori Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai nokath, bulatan atau titik, sedangkan hubungan antar objek dinyatakan dengan baris.

Graf dapat didefinisikan sebagai pasangan himpunan (V,E) dengan V menyatakan *Vertex* atau titik dan E menyatakan *Edge* atau garis yang menghubungkan sepasang titik.

$$G = (V,E)$$

2) Jenis Graf

Graf dapat dikelompokkan menjadi berbagai kategori bergantung dari dasar pengelompokkannya. Berdasarkan ada tidaknya gelang atau sisi ganda, graf dikelompokkan menjadi dua yaitu :

- Graf Sederhana
Graf sederhana adalah graf yang tidak memiliki sisi gelang maupun sisi ganda
- Graf Tak-Sederhana
Graf Tak-Sederhana adalah graf yang memiliki gelang maupun sisi ganda. Graf yang hanya memiliki sisi ganda disebut **graf ganda** dan graf yang memiliki sisi ganda maupun sisi gelang dinamakan **graf semu**

Berdasarkan jumlah simpul, graf dikelompokkan menjadi dua yaitu :

- Graf Berhingga

¹Rinaldi Munir, *Matematika Diskrit*, Informatika, Bandung, 2006, hlm. VIII

Graf berhingga adalah graf yang memiliki simpul dengan jumlah berhingga, misalnya n simpul.

- Graf Tak-Berhingga
Graf yang memiliki jumlah simpul yang tak berhingga banyaknya.

Sedangkan berdasarkan orientasi arah, graf dapat dibedakan menjadi dua yaitu :

- Graf Berarah
Graf berarah adalah graf yang sisinya diberikan orientasi arah menuju atau menjadi suatu simpul tertentu sehingga $(v_j, v_k) \neq (v_k, v_j)$.
- Graf Tak-Berarah
Graf Tak-Berarah merupakan graf yang sisinya tidak memiliki orientasi arah tertentu sehingga $(v_j, v_k) = (v_k, v_j)$ merupakan sisi yang sama.

3) Terminologi Graf

Dalam mempelajari graf, ada beberapa terminologi yang perlu dipahami, diantaranya adalah :

- Bertetangga (*Adjacent*)
Dua buah simpul pada graf tak-berarah dinyatakan bertetangga apabila kedua simpul tersebut terhubung.
- Bersisian (*Incident*)
Untuk sembarang sisi $e = (v_j, v_k)$, sisi e dikatakan bersisian dengan simpul v_j dan v_k .
- Simpul Terpencil (*Isolated Vertex*)
Suatu simpul dinyatakan sebagai simpul terpencil apabila tidak terdapat sisi yang bersisian dengan simpul tersebut.
- Graf Kosong (*Null Graph*)
Graf kosong merupakan graf yang himpunan sisinya merupakan himpunan kosong.
- Derajat (*Degree*)
Derajat suatu simpul pada graf tak-berarah dinyatakan sebagai banyaknya sisi yang bersisian dengan simpul tersebut.
- Lintasan (*Path*)
Lintasan yang panjangnya n dari simpul awal v_0 menuju simpul v_k dengan melewati berbagai sisi dan simpul secara bergantian.
- Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Siklus atau sirkuit merupakan lintasan yang berawal dan berakhir pada simpul yang sama.

- Terhubung (*Connected*)
Suatu graf tak-berarah merupakan graf terhubung jika untuk setiap simpul pada graf tersebut, terdapat lintasan yang menuju simpul tersebut.
- Upagraf (*Subgraph*)
Upagraf merupakan bagian dari suatu graf G atau dapat disebut pula upagraf merupakan subset dari suatu graf.
- Upagraf Merentang (*Spanning Subgraph*)
Suatu upagraf disebut sebagai upagraf merentang apabila pada upagraf tersebut terdapat semua simpul graf utama.
- *Cut-Set*
Cut-Set dari suatu graf adalah apabila beberapa anggota dari himpunan sisi dibuang menyebabkan graf tersebut tidak menjadi terhubung.
- Graf Berbobot (*Weighted Graph*)
Graf berbobot adalah graf yang setiap sisinya diberikan suatu nilai atau bobot.

B. Algoritma A* (A star)

1) Sejarah Algoritma A*

Algoritma A* pertama kali ditemukan pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Dalam tulisan mereka, algoritma ini dinamakan algoritma A. Penggunaan algoritma ini dengan fungsi heuristik yang tepat dapat memberikan hasil yang optimal, maka algoritma ini pun disebut algoritma A*.

2) Definisi Algoritma A*

Algoritma A* adalah algoritma pencarian graf/pohon yang mencari jalur dari satu titik awal ke sebuah titik akhir yang telah ditentukan. Algoritma A* menggunakan pendekatan heuristik $h(x)$ yang memberikan peringkat ke setiap titik x dengan cara memperkirakan rute terbaik yang dapat dilalui dari titik tersebut. Setelah itu setiap titik x tersebut dilakukan pengecekan satu-persatu berdasarkan urutan yang dibuat dengan pendekatan heuristik tersebut. Maka dari itulah algoritma A* adalah contoh dari *Best-First Search*.

Fungsi Evaluasi $f(n) = g(n) + h(n)$

$g(n)$ = jarak untuk menuju n

$h(n)$ = estimasi jarak dari n menuju *goal*

$f(n)$ = estimasi total *cost* dengan melalui n ke *goal*

3) Terminologi Dasar

- *Starting Point*
Starting Point adalah sebuah terminologi posisi awal sebuah benda
- A
A adalah simpul yang sedang dijalankan algoritma pencarian jalan terpendek.
- Simpul (*nodes*)
Simpul adalah petak-petak kecil sebagai representasi dari *area path-finding*. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.
- *Open List*
Open List adalah tempat menyimpan data simpul yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan.
- *Closed List*
Closed List adalah tempat menyimpan data simpul sebelum A yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.
- Harga (*cost*)
Harga adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke A, dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.
- Rintangan (*unwalkable*)
Rintangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dialui oleh A.

4) Pseudocode Algoritma A*

Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah *starting point* menuju simpul tujuan dengan memperhatikan harga(F) terkecil. A* memperhitungkan *cost* dari *current state* ke tujuan dengan fungsi heuristik. Algoritma ini juga mempertimbangkan *cost* yang telah ditempuh selama ini dari *initial state* ke *current state* sehingga jika ada jalan yang telah ditempuh sudah terlalu panjang dan ada kaman lain yang *cost*-nya lebih kecil tetapi memberikan posisi yang sama dilihat dari *goal*, jalan yang lebih pendek yang akan dipilih.

Algoritma A* dapat dijelaskan dengan *pseudocode* di bawah ini :

<i>pseudocode</i> Algoritma A*
OpenList = List of Simpul yang akan diekspan
CloseList = List of Simpul yang sudah diekspan

²Grobvater. A* Pathfinding (Algoritma Pencarian Rute A*) Part 1. <http://grobvater.blogspot.co.id/2011/03/pathfinding-algoritma-pencarian-rute.html>

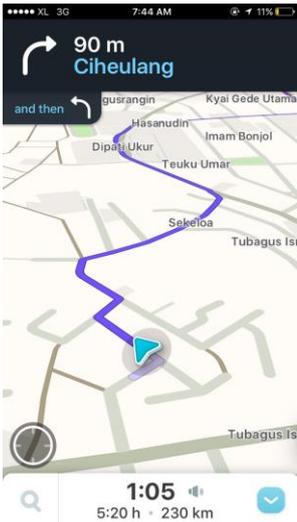
³Waze. <https://id.wikipedia.org/wiki/Waze>.

```
rute = List of Urutan jalan dari simpul awal ke simpul tujuan
Begin
While (OpenList ≠ Empty and not found solusi) do
  f_simpul_sekarang = g(simpul_sekarang) + h(simpul_sekarang)
  if (simpul_sekarang = simpul_tujuan) then
    solusi = true //solusi ditemukan
  else
    if (f_simpul_sekarang < f simpul lainnya) then
      delete simpul_sekarang from OpenList
      tambahkan simpul_sekarang ke dalam rute
      for (setiap tetangga simpul_sekarang) do
        if (simpul_tetangga tidak ada pada CloseList)
        then
          if (simpul_tetangga tidak ada di OpenList) then
            masukkan simpul_tetangga pada OpenList dan
            menjadikan simpul_sekarang sebagai parent
            dari simpul_tetangga. Kemudian hitung f(x)
            dari simpul_tetangga.
          else
            if (simpul_tetangga lebih baik dari simpul
            pada OpenList) then
              Hapus simpul dengan nilai f(x) lebih buruk
              dari nilai f(x) simpul_tetangga kemudian
              masukkan simpul_tetangga ke OpenList
              tambahkan simpul_sekarang ke CloseList
End.
```

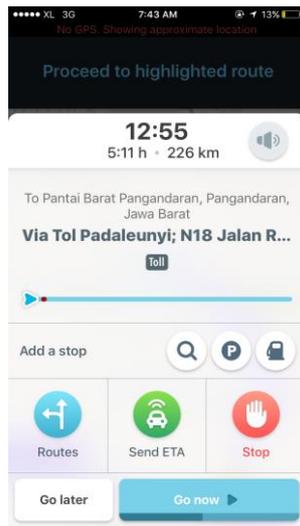
C. Aplikasi Waze³

Waze adalah sebuah aplikasi navigasi gratis untuk perangkat telepon genggam dan tablet PC yang memiliki GPS. Aplikasi *Waze* berbasis iOS, Android, Windows Mobile, Symbian, dan BlackBerry. *Waze* dapat diunduh di berbagai negara karena *Waze* telah memiliki data peta setiap negara dengan akurat. Namun, peta-peta tersebut masih memiliki berbagai kekurangan sehingga dari setiap negara memiliki komunitas *Waze* yang terus bekerja untuk menyempurnakan peta-peta guna mengikuti perkembangan dan pembangunan dari setiap negaranya.

Berbeda dengan perangkat lunal navigasi umumnya, *Waze* memberikan informasi dan peta berdasarkan masukan komunitas pemakainya. Informasi tentang kecelakaan, kemacetan jalan, polisim bahaya berdasarkan kondisi nyata yang dilaporkan penggunaannya. Pengguna *Waze* yang juga disebut *wazers* juga bisa melakukan pemutakhiran peta, penambahan tempat dan jalan baru, termasuk mengkondisikan aturan dan alur lalu lintas di setiap jalan dan persimpangannya. *Waze* juga mempunyai fasilitas ngobrol (*chat*), memberikan poin untuk setiap kegiatan yang dilakukan seperti menjelajah, memutakhirkan peta dan epristiwa khusus lainnya. Dengan demikian *Waze* adalah gabungan dari aplikasi navigasi dengan jejaring sosial dan permainan *online*.



Gambar 2 : Jalan dengan Kemampuan Laju Kecepatan Tertentu
Sumber : Koleksi Penulis



Gambar 1 : Aplikasi Waze
Sumber : Koleksi Penulis

Dalam menentukan jalan terpendek dan waktu tempuh terkecil, peta *Waze* dibutuhkan dengan menganggap peta tersebut sebagai suatu graf. Agar peta *Waze* dapat dianggap sebagai suatu graf, setiap persimpangan menuju nama jalan lainnya dianggap sebagai simpul dan jalan yang menghubungkan setiap persimpangan tersebut dianggap sebagai sisi.

Pada Gambar 2 terlihat bahwa suatu kendaraan, terutama mobil, dapat melaju dengan kecepatan tertentu pada jalur yang padat kendaraan atau mengalami kemacetan. Dengan adanya informasi tersebut, fungsi heuristik algoritma A* dalam menentukan jalan terpendek dengan waktu tempuh terkecil dapat ditentukan.

Fungsi heuristik pada aplikasi *Waze* dapat ditentukan dengan cara sebagai berikut. Setiap informasi laju yang diperbarui oleh para **wazers** akan dikalkulasi menjadi rumus jarak. Kalkulasi tersebut ditentukan dengan rumus sebagai berikut

$$\text{Jarak (S)} = \text{Kecepatan (V)} * \text{Waktu (T)}.$$

Dengan persamaan tersebut, akan dicari waktu tempuh dari setiap jalan dengan laju tertentu. Apabila tidak ada informasi kemacetan pada suatu jalan dari para **wazers**, maka kecepatan normal pengemudi diasumsikan adalah 60km/jam. Dengan demikian, penentuan jalur menuju lokasi tujuan adalah dengan fungsi waktu tempuh dan menyatakan ketiga fungsi pada algoritma A* sebagai berikut

- g(x) = waktu tempuh menuju suatu persimpangan
- h(x) = estimasi waktu dari suatu persimpangan menuju lokasi tujuan
- f(x) = estimasi waktu total dengan melalui persimpangan tertentu menuju lokasi tujuan

III. ANALISIS ALGORITMA A* PADA APLIKASI WAZE

A. Fungsi Heuristik

Aplikasi *Waze* memiliki *Waze* API sendiri untuk menampilkan peta pada aplikasi tersebut. Peta tersebut dikembangkan oleh para **wazers** atau komunitas *Waze* untuk melakukan perkembangan mengenai kondisi jalan yang terbaru. Para **wazers** dapat memberi tahu pengguna *Waze* lainnya mengenai kecepatan yang dapat dilaju pada suatu jalan. Sehingga informasi dari para **wazers** ini dapat dimanfaatkan sebagai fungsi heuristik dalam menentukan jalan dengan jalur terpendek dengan waktu tempuh terkecil.

sebagai contoh, pada gambar 2, fungsi heuristik untuk menuju persimpangan dago melalui jalan Tubagus Ismail Raya dengan *start point* adalah dari depan jalan Tubagus Ismail VII adalah

$$6\text{km/jam} = 1,6\text{m/s}$$

Jarak dari depan jalan Tubagus Ismail VII menuju persimpangan dago melalui jalan Tubagus Ismail Raya adalah 630 meter

Sehingga didapat waktu tempuh

$$\text{Waktu (T)} = 630\text{m} / (1,6\text{m/s}) = 393,75\text{s} = 6,5625\text{menit} = 0,2\text{jam}$$

B. Pseudocode Penerapan Algoritma A* pada Aplikasi Waze
 Pseudocode Algoritma A* aplikasi Waze dapat dijabarkan sebagai berikut :

1. Menentukan *start point*
2. Memasukkan simpul awal ke OpenList
3. Memilih simpul tetangga dan menghitung nilai $f(x)$ simpul tersebut dengan $f(x) = g(x) + h(x)$
4. Nilai $h(x)$ suatu simpul dikalkulasi dengan cara sebagai berikut
 - Mendapatkan waktu yang akan ditempuh dari suatu titik ke titik lainnya dengan mengansumsikan kecepatan yang tidak diberikan informasinya oleh para **wazers** adalah 60km/jam.
 - Mengkonversi kecepatan yang diberitahukan oleh para **wazers** atau kecepatan asumsi ke dalam m/s
 - Menghitung waktu tempuh dari suatu tempat ke tempat lainnya dengan rumus kecepatan yaitu jarak = kecepatan * waktu. Waktu yang didapatkan dalam sekon.
5. Melakukan pengulangan untuk setiap simpul pada OpenList dan melakukan langkah berikut
 - Mencari simpul dengan nilai $f(x)$ terkecil pada OpenList
 - Untuk setiap tetangga pada dari simpul dengan $f(x)$ terkecil tersebut dilakukan :
 - Jika simpul tetangga dari simpul tersebut sudah ada pada CloseList maka simpul tersebut diabaikan
 - Jika simpul tetangga tersebut **belum** ada pada OpenList, buat simpul yang sedang diekspan menjadi simpul *parent* dari simpul tetangga tersebut dan menyimpan nilai f , g , dan h simpul tetangga tersebut.
 - Jika simpul tetangga tersebut **sudah** ada pada OpenList, cek jika simpul tetangga tersebut lebih baik, menggunakan nilai g sebagai ukuran. Jika lebih baik, maka ganti *parent* dari simpul tetangga ini dengan simpul sekarang, lalu lakukan

kalkulasi ulang nilai f , g , dan h dari simpul ini.

- Pengulangan dihentikan apabila simpul tujuan telah ditemukan dan OpenList belum kosong.
6. Menyimpan rute secara *backward* urut mulai dari simpul tujuan ke *parent*-nya hingga mencapai simpul awal sambil menyimpan simpul-simpul tersebut ke dalam sebuah *List* atau *Array*.

```

pseudocode Algoritma A*

function h ()
//Kamus Lokal
integer waktu

//Algoritma Lokal
kecepatan = kecepatan * 1000 / 3600
waktu = kecepatan / jarak pada Peta

return waktu

//Main Program

OpenList = List of Simpul yang akan diekspan
CloseList = List of Simpul yang sudah diekspan
rute = List of Urutan jalan dari simpul awal ke simpul tujuan
Begin
    tambahkan simpul awal ke OpenList
    While (OpenList ≠ Empty and not found solusi) do
        f_simpul_sekarang = g(simpul_sekarang) + h(simpul_sekarang)
        if (simpul_sekarang = simpul_tujuan) then
            solusi = true //solusi ditemukan
        else
            if (f_simpul_sekarang < f simpul lainnya) then
                delete simpul_sekarang from OpenList
                tambahkan simpul_sekarang ke dalam rute
                for (setiap tetangga simpul_sekarang) do
                    if (simpul_tetangga tidak ada pada CloseList)
                then
                    if (simpul_tetangga tidak ada di OpenList) then
                        masukkan simpul_tetangga pada OpenList dan
                        menjadikan simpul_sekarang sebagai parent
                        dari simpul_tetangga. Kemudian hitung f(x)
                        dari simpul_tetangga.
                    else
                        if (simpul_tetangga lebih baik dari simpul
                        pada OpenList) then
                            Hapus simpul dengan nilai f(x) lebih buruk
                            dari nilai f(x) simpul_tetangga kemudian
                            masukkan simpul_tetangga ke OpenList
                            tambahkan simpul_sekarang ke CloseList
            End.
    
```

IV. KESIMPULAN

Algoritma A* (A star) dapat digunakan untuk menyelesaikan berbagai persoalan di antaranya adalah navigasi pada suatu aplikasi penunjuk jalan dari suatu lokasi awal menuju lokasi tujuan. Penerapan Algoritma A* pada aplikasi tersebut memanfaatkan fungsi heuristik waktu tempuh dari suatu tempat ke tempat lainnya. Hasil dari penerapan algoritma A* ini merupakan solusi optimal dengan waktu tempuh terkecil dan diharapkan dengan jarak terpendek apabila kecepatan yang dapat dilaju pada setiap ruas jalan adalah sama. Apabila kecepatan yang dapat dilaju pada setiap ruas jalan berbeda-beda, maka kemungkinan untuk mendapatkan petunjuk jalan dengan jalan terpendek dan waktu tempuh terkecil menjadi lebih sulit dan aplikasi akan menunjukkan jalan dengan waktu tempuh terkecil saja.

UCAPAN TERIMA KASIH

Pertama penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa karena dengan rahmat dan izinnya penulis dapat menyelesaikan makalah *technical report* dengan judul “Penerapan Algoritma A* (A star) pada Aplikasi Waze” ini dengan baik. Terimakasih juga saya ucapkan kepada dosen pengampu mata kuliah IF2211 Strategi Algoritma yaitu ibu Dr.Nur Ulfa Maulidevi, bapak Dr. Ir. Rinaldi Munir MR, dan ibu Masayu Leyla Khodra ST., MT. atas bimbingan dalam mengajarkan saya mata kuliah Strategi Algoritma. Semoga segala ilmu yang telah diberikan bapak dan ibu akan menjadi ilmu yang bermanfaat bagi penulis dan dapat penulis terapkan bagi kebutuhan orang banyak.

REFERENSI

- [1] Munir, Rinaldi. 2006. *Matematika Diskrit*. Bandung : Informatika
- [2] “Blog”. <https://www.waze.com/>. Diakses pada tanggal 19 Mei 2017.
- [3] Rinaldi Munir, *Route/ Path Planning Using A Star and UCS*. 2016. [Online] Tersedia dalam <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/stima15-16.htm>. Diakses pada tanggal 19 Mei 2017.
- [4] Gribvater. *A* Pathfinding (Algoritma Pencarian Rute A*) Part 1*. <http://grobvater.blogspot.co.id/2011/03/pathfinding-algoritma-pencarian-rute.html>. Diakses pada tanggal 19 Mei 2017.
- [5] Waze. <https://id.wikipedia.org/wiki/Waze>. Diakses pada tanggal 19 Mei 2017
- [6] *Sejarah*. <https://id.wikipedia.org/wiki/Waze>. Diakses pada tanggal 19 Mei 2017.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Mei 2017



Aya Aurora Rimbamorani 13515098