

Metode Pencarian Lintasan Terpendek Graf untuk Evakuasi Bencana

Hanif Lyonais (13510108)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13510108@std.stei.itb.ac.id

Abstract—Makalah ini membahas penerapan algoritma untuk membantu evakuasi bencana. Evakuasi bencana merupakan salah satu permasalahan yang terjadi di negara ini. Kondisi alam negara Indonesia memang cukup berpotensi menyebabkan terjadinya bencana alam. Saat terjadi bencana, evakuasi korban adalah elemen penting yang harus diperhatikan, tentu saja evakuasi ini harus mangkus.

Algoritma A* dan Dijkstra merupakan salah satu algoritma komputer yang sudah cukup populer digunakan. Salah satu penggunaannya yaitu *pathfinding*, mencari jalan. Algoritma A* dan Dijkstra menyelesaikan masalah yang direpresentasikan menggunakan graf. Algoritma ini mencari rute terpendek dari node awal hingga node tujuan pada graf. Dengan memanfaatkan algoritma ini kita dapat mencari rute terdekat menuju tempat yang aman saat evakuasi bencana.

Kata Kunci—algoritma A*, bencana, evakuasi, rute terpendek

I. PENDAHULUAN

Indonesia merupakan salah satu wilayah yang rawan bencana. Baik bencana alam maupun bencana yang diakibatkan oleh ulah manusia sendiri. Kondisi alam yang berubah – ubah bisa menyebabkan banjir, tanah longsor, gempa bumi, gunung meletus, bahkan tsunami. Kerugian materiil yang diakibatkan tidak sedikit akan tetapi tentu saja korban jiwa adalah kerugian yang tidak ternilai. Seiring dengan kemajuan zaman, kemampuan manusia untuk menyesuaikan diri dengan lingkungannya semakin meningkat. Menanggapi kondisi – kondisi yang membahayakan, manusia membuat mekanisme mitigasi bencana. Salah satu bentuk mitigasi bencana tersebut yaitu dengan evakuasi.

Evakuasi adalah pengungsian atau pemindahan penduduk dari daerah – daerah yang berbahaya, misal lokasi banjir, kaki gunung merapi, daerah pantai saat tsunami, dan lain –lain. Saat terjadi bencana tentu saja daerah – daerah sekitar pusat bencana harus di jauhi. Warga yang terdapat di sekitar daerah berbahaya harus segera dipindahkan ke tempat yang aman. Sistem kounikasi memainkan perna penting dalam keselamatan korban. Biasanya digunakan sistem komunikasi audio seperti alarm untuk menyampaikan peringatan terjadinya bencana

Pada saat evakuasi harus diperhitungkan jumlah warga yang dimobilisasi, kapasitas orang yang bisa dibawa, serta rute yang dipilih menuju tempat aman. Dengan waktu yang sempit faktor – faktor tersebut harus diperhitungkan demi keselamatan semua orang.

Bagian yang cukup penting dalam evakuasi yaitu memilih rute menuju tempat aman. Warga yang dimobilisasi harus diantar ke tempat yang aman melalui rute terpendek. Evakuasi harus dilakukan semangkus mungkin, waktu yang dipergunakan untuk memindahkan warga harus seminimal mungkin, hambatan dalam rute harus dihindari.

Pemilihan rute dalam evakuasi ini memperhimbangkan jalan – jalan yang ada serta berdasarkan data geografis dari daerah bencana itu sendiri. Tiap daerah yang terimpa bencana memiliki karakteristik yang berbeda – beda. Oleh karena itu untuk menyelesaikan masalah rute ini bisa diperlukan metode tertentu. Kita bisa menggunakan Algoritma A* dan Dijkstra untuk menentukan rute terpendek.

Algoritma *A ini berdasarkan pada graf. Dimulai dari sebuah *starting point*, yaitu titik awal, kemudian ada *node*, node ini bisa diibaratkan titik – titik sebagai representasi dari daerah yang dilalui. Ada juga harga(*cost*), Harga (*cost*) adalah nilai yang diperoleh dari penjumlahan *nilai G* (jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke A), dan *nilai H* (jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan).

Dalam pencarian rute ini juga terdapat atribut rintangan, yaitu sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A.

Jadi prinsip algoritma *A ini adalah mencari jalur terpendek dari sebuah simpul awal (*starting point*) menuju simpul tujuan dengan memperhatikan harga (*cost*).

Algoritma Dijkstra, adalah sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah grafberarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif.

Algoritma – algoritma tersebut diintegrasikan dengan aplikasi peta digital seperti ArcGIS sehingga kita bisa menentukan rute dari tempat terjadinya bencana ke tempat aman.

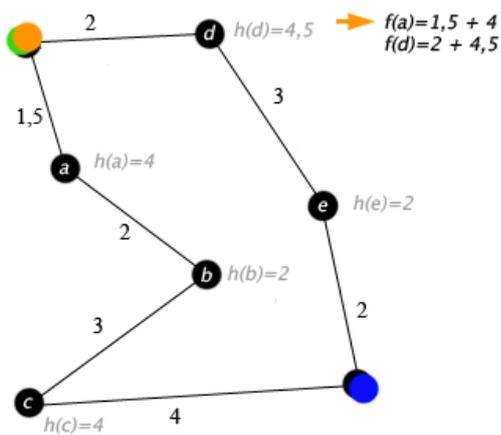
II. ALGORITMA A*

Pada tahun 1964 Nil Nilsson menciptakan pendekatan berbasis heuristik untuk meningkatkan kecepatan algoritma Dijkstra. Algoritma ini disebut A1. Pada tahun 1967 Bertram Raphael membuat perbaikan dramatis pada algoritma ini, tapi algoritma ini tidak bekerja optimal. Dia menyebut algoritma ini A2. Kemudian pada 1968 Peter E.Hart memperkenalkan sebuah argumen yang membuktikan A2 akan optimal bila menggunakan heuristik yang konsisten dan hanya dengan perubahan kecil. Pembuktian algoritma ini juga termasuk menunjukkan bahwa algoritma A2 baru adalah algoritma terbaik sesuai kondisinya. Kemudian ia menamai algoritma ini A*.

Pada saat memilih jalan, algoritma A* menghitung semua kemungkinan jalan yang ada kemudian kemungkinan tersebut disimpan. Setelah itu ia membandingkan dengan hasil jalan lain yang telah disimpan. Oleh karena itu hasil pencarian jalan dengan algoritma ini menghasilkan hasil yang optimum.

Akan tetapi karena terus membandingkan jalan yang dicari, algoritma ini memakan cukup banyak waktu. Jika simpulnya banyak maka pencarian pun akan semakin lama.

Di bawah ini penjelasan cara kerja algoritma A* :



Gambar 1. Graf

Kita akan mencari rute jalan terpendek dari titik kuning menuju ke titik biru.

Pertama- tama kita lihat jalurnya

- Rute1 (a b) = 1,5
- Rute2 (d) = 2
- Dipilih Rute1 karena lebih pendek
- Rute1 (a b) = 3,5
- Rute2 (d) = 2
- Dipilih Rute 2 karena lebih pendek

- Rute1 (a b) = 3,5
- Rute2 (d e) = 5
- Dipilih Rute 1 karena lebih pendek

- Rute1 (a b c) = 6,5
- Rute2 (d e) = 5
- Dipilih Rute 2 karena lebih pendek

- Rute1 (a b c) = 6,5
- Rute2 (d e) = 7

Meskipun telah sampai tujuan tetapi jalur lain belum mencapai tujuan sehingga pencarian masih dilanjutkan

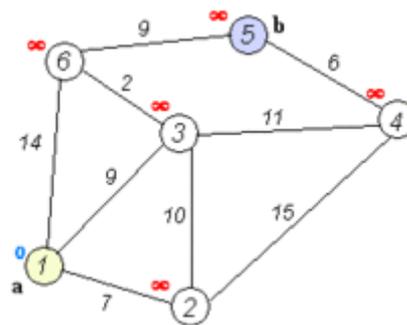
Jarak1 (a b c) = 10,5

Jarak2 (d e) = 7

Setelah semua jalur telah tercapai baru dibandingkan kembali.

Karena jalur 2 lebih pendek maka dipilihlah Rute 2.

Untuk melakukan pencarian pada graf yang kecil seperti **Gambar 1** algoritma ini sudah memakan waktu yang sangat banyak. Padahal graf tersebut hanya memiliki 2 jalur. Bayangkan jika graf tersebut sedikit lebih rumit seperti di bawah ini



Pseudocode untuk algoritma A*

```
function A*(start, goal)
    closedset := the empty set //
    The set of nodes already evaluated.
    openset := {start} // The set
    of tentative nodes to be evaluated,
    initially containing the start node
    came_from := the empty map //
    The map of navigated nodes.

    g_score[start] := 0 // Cost
    from start along best known path.
    h_score[start] :=
    heuristic_cost_estimate(start, goal)
    f_score[start] := g_score[start]
    + h_score[start] // Estimated total
    cost from start to goal through y.
```

```

    while openset is not empty
        x := the node in openset
        having the lowest f_score[] value
        if x = goal
            return
    reconstruct_path(came_from,
    came_from[goal])
    remove x from openset
        add x to closedset
        foreach y in
neighbor_nodes(x)
            if y in closedset
                continue
            tentative_g_score

            if y not in openset
                add y to openset

tentative_is_better := true
else if tentative_g_score <
g_score[y]
tentative_is_better := true
else
tentative_is_better := false
if tentative_is_better = true
    came_from[y] := x
    g_score[y] :=
tentative_g_score
    h_score[y] :=
heuristic_cost_estimate(y, goal)
    f_score[y] :=
g_score[y] + h_score[y]

    return failure

function reconstruct_path
(came_from, current_node)
    if came_from[current_node] is
set
        p :=
reconstruct_path(came_from,
came_from[current_node])
        return (p + current_node)
    else
        return current_node

```

Waktu yang diperlukan untuk mencari rute terpendek dengan algoritma A* cukup lama.

Kompleksitas Waktu Asimptotik dari algoritma a* Adalah

$$|H(X) - H^*(X)| = O(\text{LOG}H^*(X))$$

III. ALGORITMA DIJKSTRA

Algoritma lain yang cukup terkenal untuk menyelesaikan persoalan ini adalah algoritma Dijkstra. Algoritma ini ditemukan oleh seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra.

Dijkstra adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Contoh penerapan algoritma dijkstra adalah lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu (Single-source Singledestination Shortest Path Problems). Cara kerja algoritma Dijkstra memakai strategi greedy, di mana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal, dan tempat tujuan. Hasil akhir dari algoritma ini adalah jarak terpendek dari tempat asal ke tempat tujuan beserta rutenya.

Pseudocode Algoritma Dijkstra

```

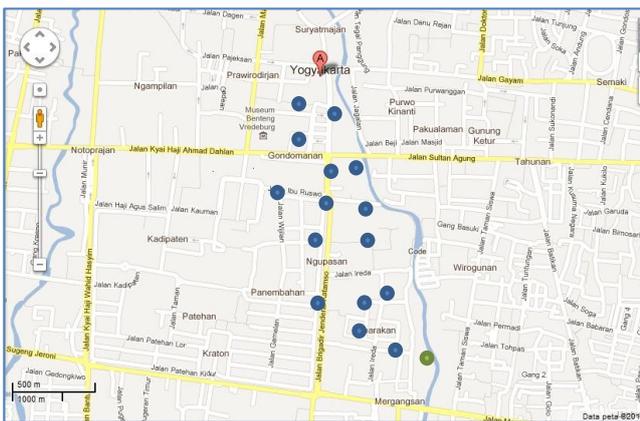
function Dijkstra(G, w, s)
    for each vertex v in V[G]           // Initializations
        d[v] := infinity
        previous[v] := undefined
    d[s] := 0                             // Distance from s to s
    S := empty set
    Q := V[G]                             // Set of all vertices
    while Q is not an empty set         // The
algorithm itself
        u := Extract_Min(Q)
        S := S union {u}
        for each edge (u,v) outgoing from u
            if d[u] + w(u,v) < d[v]     // Relax (u,v)

```

IV. PENERAPAN PADA PETA

ArcGIS

ArcGIS adalah sebuah perangkat lunak yang memungkinkan seseorang untuk melihat peta membuat peta, melakukan beberapa analisis spasial. Hal ini juga memungkinkan penggunaan data pada peta (seperti jalan raya, jenis zona dan polusi peta terkait) dimana yang satu dapat bersinggungan dengan yang lain.. Untuk tujuan proyek, file yang diproses di ArcGIS dapat digunakan untuk mengevaluasi data. Dengan menggunakan ArcGIS kita dapat menentukan titik – titik daerah aman dari bencana. Kemudian dengan menggunakan *pathfinding* dengan algoritma A* dan Dijkstra kita dapat rute tercepat menuju titik tersebut.



Gambar 2

Dari gambar di atas bisa kita lihat titik – titik yang merepresentasikan daerah yang aman. Di peta starting point nya adalah titik merah dan daerah aman yang dituju yaitu titik hijau. Kita ingin dengan menggunakan algoritma – algoritma yang dijelaskan sebelumnya kita dapat menentukan rute tercepat untuk evakuasi. Ini sangat penting mengingat pada saat terjadi bencana, waktu adalah hal yang sangat berharga dan semua upaya yang dilakukan harus semangkus mungkin.

Dengan menggunakan algoritma A* kita dapat menambahkan berbagai variabel dalam penentuan rute tercepat. Pada bagian heuristik kita dapat menyertakan kecepatan mobil, keramaian jalan, dll

Dengan menggunakan algoritma Dijkstra kita hanya menentukan rute jalan tercepat saja.

V. KESIMPULAN

Penentuan rute jalan terpendek merupakan salah satu aplikasi terori graf.

Penggunaan algoritma A* dan Dijkstra sama – sama

dapat dipergunakan untuk menentukan rute jalan terpendek.

Dalam algoritma A* ada fungsi heuristik dimana bisa dimasukkan berbagai variabel dalam penentuan rute terpendek

Penentuan rute kecepatan dengan Algoritma A* menghabiskan lebih banyak waktu dibanding dengan algoritma Dijkstra, tetapi rute yang dihasilkan algoritma A* lebih optimum.

REFERENSI

- [1] http://en.wikipedia.org/wiki/A*_search_algorithm waktu akses : 11 Desember 2011
- [2] <http://www.policymanac.org/games/aStarTutorial.htm> waktu akses : 11 Desember 2011
- [3] http://id.wikipedia.org/wiki/Algoritma_Dijkstra waktu akses : 11 Desember 2011
- [4] Rinaldi, Munir, *Matematika Diskrit edisi III*. Bandung: Informatika Bandung, 2007.

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, 12 Desember 2012



Hanif Lyonnais