



IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE TERPENDEK TEMPAT WISATA DI KABUPATEN GUNUNGKIDUL DENGAN PROGRAM VISUAL BASIC

Stepanus Ardyan, Mulyono, Amin Suyitno

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia
Gedung D7 Lt.1, Kampus Sekaran Gunungpati, Semarang 50299

Info Artikel

Sejarah Artikel:

Diterima Agustus 2016

Disetujui Juli 2017

Dipublikasikan November 2017

Keywords:

Rute Terpendek;

Algoritma Dijkstra;

Visual Basic

Abstrak

Algoritma Dijkstra dapat digunakan untuk mencari rute terpendek dari sebuah titik yang ditentukan ke semua titik dalam graf. Kabupaten Gunungkidul merupakan salah satu Kabupaten yang memiliki banyak tempat wisata yang tersebar. Tempat wisata yang tersebar tersebut menyebabkan para wisatawan sulit menentukan rute yang harus dilalui agar dapat menikmati beberapa tempat wisata yang berbeda. Setiap orang yang melakukan perjalanan pasti memilih rute terpendek untuk dapat mencapai tujuan karena dapat menghemat waktu, tenaga dan biaya bahan bakar. Ketika berwisata dengan jadwal yang tidak diatur menyebabkan pengeluaran anggaran berwisata membesar dan waktu berlibur menjadi padat. Dari permasalahan tersebut maka penulis menganalisis rute terpendek tempat wisata di Kabupaten Gunungkidul dengan algoritma Dijkstra dan membuat simulasi pencarian rute terpendek pada graf tak berarah dan berbobot dengan bahasa pemrograman Visual Basic sehingga dapat menghemat biaya dan waktu wisatawan yang berwisata ke Kabupaten Gunungkidul.

Abstract

Dijkstra algorithm can be used to find the shortest route from a node to another node in a graph. Gunungkidul Regency is one of districts that has many tourist spots are scattered. The tourist spot which scattered make the tourist have difficulty to specify the route to be followed in order to be able to enjoy several tourist spots. Everyone who travels certainly choose the shortest route to reach the goal because it can save time, costs, energy, and fuel. When we was travelling with the schedule that is not regulated lead the budget increased and holiday become crowded. Of these problems, the author analyzed the shortest route tourist spots in Gunungkidul regency with Dijkstra's algorithm and simulated the shortest route search in undirected and weighted graph with Visual Basic programming language so as to save tourist costs and time who traveled to the Gunungkidul regency.

How to Cite

Ardyan, S., Mulyono & Suyitno, A. (2017). Implementasi Algoritma Dijkstra dalam Pencarian Rute Terpendek Tempat Wisata di Kabupaten Gunungkidul dengan Program Visual Basic. *Unnes Journal of Mathematics*, 6(2): 108-116.

PENDAHULUAN

Algoritma dapat didefinisikan sebagai urutan langkah-langkah logis dan sistematis dalam mencari suatu solusi dari suatu permasalahan yang ada. Langkah-langkah dalam memecahkan masalah bisa dilakukan dalam berbagai cara dengan karakteristik yang berbeda-beda dari masing-masing langkah. Tiap-tiap algoritma tersebut memiliki cara kerja yang berbeda-beda dalam menentukan solusi yang paling optimal. Untuk mencari panjang rute terpendek dari sebuah titik s ke sebuah titik t di graf bobot G , dengan bobot setiap sisi G adalah bilangan positif, digunakan algoritma Dijkstra.

Kabupaten Gunungkidul adalah salah satu kabupaten yang ada di Daerah Istimewa Yogyakarta, dengan Ibukotanya Wonosari. Kabupaten Gunungkidul memiliki wilayah geografis berupa daerah pegunungan dan berbatasan langsung dengan laut selatan. Kondisi tersebut membuat Gunungkidul dianugerahi oleh beragam objek wisata alam yang tersebar di hampir semua kecamatan di Gunungkidul.

Objek Wisata yang berlimpah tersebut membuat Gunungkidul menjadi tujuan wisata. Hal ini didukung dengan pembangunan akses jalan menuju objek wisata yang ada. Wisatawan yang datang bukan hanya dari Yogyakarta, namun juga dari lintas propinsi seperti dari Jakarta, Semarang, dan banyak kota lainnya. Apalagi ketika libur panjang tiba, hampir semua objek wisata penuh oleh wisatawan.

Istilah Riset Operasi pertama kali digunakan pada tahun 1940 oleh Mc Closky dan Trefthen di suatu kota kecil, Bowdsey, Inggris. Pada masa awal perang 1939, pemimpin militer Inggris memanggil sekelompok ahli-ahli sipil dari berbagai disiplin dan megkoordinasikan mereka ke dalam suatu kelompok yang disertai tugas mencari cara-cara yang efisien untuk menggunakan alat yang baru ditemukan yang dinamakan radar dalam suatu sistem peringatan dini menghadapi serangan udara. Kelompok ahli Inggris ini dan kelompok-kelompok lain berikutnya melakukan penelitian (*research*) pada operasi-operasi (*operations*) militer. Hasilnya sangat memuaskan, kesuksesan proyek manajemen radar ini menyebabkan pemimpin militer lebih mengandalkan riset operasi dalam membuat suatu keputusan operasional yang penting (Hilier and Lieberman, 1990: 4).

Riset operasi merupakan pengambilan keputusan dengan memanfaatkan pengetahuan ilmiah melalui usaha kelompok antar disiplin yang bertujuan untuk menentukan penggunaan terbaik sumber daya yang terbatas. Model riset operasi berkaitan dengan data deterministik biasanya jauh lebih sederhana dari pada yang

melibatkan data probabilitistik (Taha, 1997: 4). Riset Operasi, dalam arti luas dapat diartikan sebagai penerapan metode- metode, teknik-teknik dan alat-alat terhadap masalah-masalah yang menyangkut operasi-operasi dari sistem-sistem, sedemikian rupa sehingga memberikan penyelesaian optimal (Mulyono, 2004 : 4).

Optimasi adalah salah satu disiplin ilmu dalam matematika yang fokus untuk mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang, maupun pencarian nilai lainya dalam berbagai kasus. Optimasi sangat berguna di hampir segala bidang terutama bidang induatri dalam rangka melakukan usaha secara efektif dan efisien untuk mencapai target hasil yang ingin dicapai. Tentunya hal ini akan sangat sesuai dengan prinsip ekonomi yang berorientasikan untuk senantiasa menekan pengeluaran untuk menghasilkan outputan yang maksimal. Optimasi ini juga penting karena persaingan saat ini sudah benar benar sangat ketat (Pradana, 2009).

Pencarian rute terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian rute terpendek ini telah diterapkan di berbagai bidang untuk mengoptimasi kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses. Salah satu aplikasi pencarian rute terpendek yang paling menarik untuk dibahas adalah pada masalah transportasi (Sulindawati, 2015: 1). Oleh karena itu, diperlukan adanya program pendukung dalam melakukan pencarian rute terpendek pada graf untuk mempercepat pencarian. Program yang dirancang adalah berupa simulasi pencarian rute terpendek menuju objek wisata yang terdapat di Kabupaten Gunungkidul. Simulasi tersebut bertujuan untuk memberi kemudahan bagi wisatawan yang ingin mengunjungi suatu objek wisata. Simulasi pencarian rute terpendek ini dibangun menggunakan bahasa pemrograman Visual Basic.

Microsoft Visual Studio .net merupakan salah satu software buatan Microsoft Corp. yang didesain khusus dalam pembuatan program-program profesional berbasis windows platform. Microsoft Visual Studio.net merupakan perangkat lunak yang terintegrasi, didalamnya terdapat beberapa paket *software* yang dapat digunakan oleh *programmer* dalam membangun sebuah program profesional salah satunya adalah Visual Basic (Rahadian, 2011: 1).

Pemrograman Visual Basic adalah suatu pemrograman visual, di mana pembuatan program dilakukan menggunakan media visual

yang salah satunya disebut dengan *user-interface* (Setyadi, 2011: 10).

Pemrograman visual merupakan dimensi baru dalam pembuatan aplikasi karena dapat langsung menggambarkan objek-objek ke layar sebelum dieksekusi. Dalam lingkungan pengembangan visual, sekarang objek yang dibuat hasilnya langsung tampil di layar. Objek yang dibuat akan sama hasilnya pada saat program dijalankan. Dengan demikian tidak perlu lagi melakukan perubahan kode program secara manual. Setelah semua objek diletakkan dalam suatu form, maka semua atribut objek tersebut akan disimpan dalam suatu kode program yang dapat langsung dijalankan (Suparno, 2011).

Salah satu keunggulan Visual Basic adalah dapat berinteraksi dengan aplikasi lain di dalam sistem operasi *Windows*, memungkinkan pengguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi *Windows* (Ricky, dkk, 2013: 141).

Permasalahan dalam penelitian ini adalah (1) Bagaimana penerapan algoritma Dijkstra dalam mencari rute terpendek pada pencarian objek wisata di Kabupaten Gunungkidul? (2) Bagaimana membangun simulasi algoritma Dijkstra dalam mencari rute terpendek pada tempat wisata di Kabupaten Gunungkidul menggunakan bahasa pemrograman Visual Basic?.

METODE PENELITIAN

Pada penelitian ini data yang digunakan adalah data sekunder yang didapat dari Google Maps yaitu data jarak antar objek wisata di Kabupaten Gunungkidul dengan batasan jalan yang dilalui adalah jalan yang dapat dilalui kendaraan roda empat secara berdampingan.

Jika riset operasi akan digunakan untuk memecahkan suatu permasalahan, maka harus dilakukan lima langkah sebagai berikut.

1. Memformulasikan persoalan.
2. Mengobservasi sistem.
3. Memformulasikan model matematis dari persoalan yang dihadapi.
4. Mengevaluasi model dan menggunakannya untuk prediksi.
5. Mengimplementasikan hasil studi (Dimiyati, 1999: 4).

Langkah-langkah untuk mencari rute terpendek dan membangun simulasi dari Visual Basic dapat dilakukan secara bertahap sebagai berikut.

1. Pengambilan data.
2. Menghitung rute terpendek dengan algoritma Dijkstra.
3. Merancang simulasi.

4. Membuat simulasi dengan bahasa pemrograman *Visual Basic*.
5. Program simulasi yang sudah dirancang kemudian diimplementasikan terhadap data yang diperoleh.
6. Mengevaluasi simulasi.
7. Kesimpulan.

Langkah-langkah algoritma Dijkstra adalah sebagai berikut.

- Input : Graf bobot G dengan $s, t \in V(G)$.
- Step 1 : Label titik s dengan $\lambda(s) = 0$ dan untuk setiap titik v di G selain s , label titik v dengan $\lambda(v) = \infty$. (dalam praktik diganti dengan bilangan yang sangat besar). Tulis $T = V(G)$.
- Step 2 : Misalkan $u \in T$ dengan $\lambda(u)$ minimum.
- Step 3 : Jika $u \in t$, berhenti, berarti panjang lintasan terpendek dari s ke t adalah $\lambda(t)$.
- Step 4 : Untuk setiap sisi $e = uv, v \in T$; ganti label v dengan $\lambda(v) = \min\{\lambda(v), \lambda(u) + w(e)\}$.
- Step 5 : Tulis $T = T - \{u\}$, dan kembali ke step 2. (Budayasa, 2007).

Prinsip kerja Algoritma Dijkstra menggunakan prinsip *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih (Diana, 2011 : 28).

Algoritma Dijkstra juga dapat digunakan untuk mencari lintasan terpendek dari sebuah titik yang ditentukan ke semua titik dalam gambar pada saat yang bersamaan, oleh sebab itu masalah tersebut seringkali disebut dengan *single-source shortest paths problem* (Kartika & Jeffrey, 2002: 69).

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 (Dewi, 2010: 47).

HASIL DAN PEMBAHASAN

Hasil Penelitian

Dalam penelitian ini, tempat wisata diasumsikan sebagai titik (*node*), dan jarak antar tempat wisata diasumsikan sebagai sisi (*edge*) yang memiliki bobot. Hasil penelitian yang telah diperoleh berupa nama tempat wisata dapat dilihat pada Tabel 1.

Tabel 1. Daftar nama tempat wisata

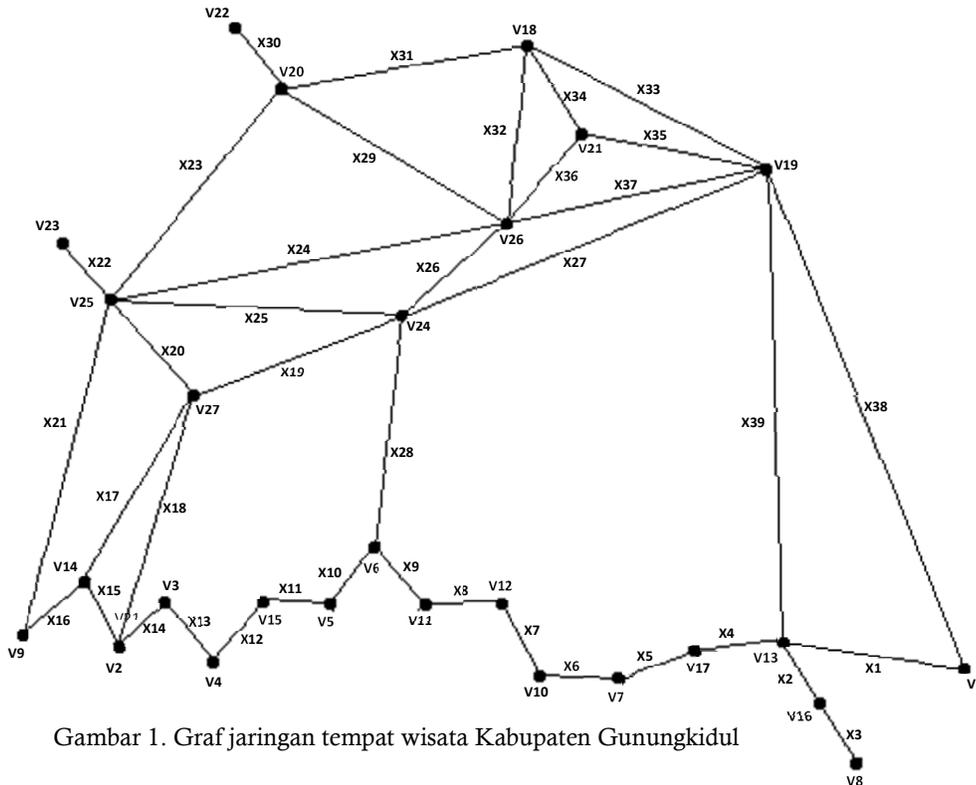
Nama Tempat Wisata	v_i
Pantai Sadeng	v_1
Pantai Baron	v_2
Pantai Kukup	v_3
Pantai Sepanjang	v_4
Pantai Krakal	v_5
Pantai Sundak	v_6
Pantai Siung	v_7
Pantai Jungwok	v_8
Pantai Ngobaran	v_9
Pantai Timang	v_{10}
Pantai Pulang Sawal	v_{11}
Pantai Pok Tunggal	v_{12}
Pantai Wediombo	v_{13}
Pantai Ngrenehan	v_{14}
Pantai Drini	v_{15}
Pantai Greweng	v_{16}
Pantai Watu Lumbung	v_{17}
Desa Wisata Wonosadi	v_{18}
Desa Wisata Umbulrejo	v_{19}
Desa Wisata Bobung	v_{20}
Desa Wisata Garotan	v_{21}
Gunung Nglanggeran	v_{22}
Air Terjun Srigethuk	v_{23}
Kalisuci	v_{24}
Goa Rancang Kencono	v_{25}
Goa Pindul	v_{26}
Gua Maria Tritis	v_{27}

$x_{15} : v_2-v_{14}$	10174
$x_{16} : v_{14}-v_9$	1234
$x_{17} : v_{14}-v_{27}$	9427
$x_{18} : v_2-v_{27}$	7138
$x_{19} : v_{27}-v_{24}$	20714
$x_{20} : v_{27}-v_{25}$	25241
$x_{21} : v_9-v_{25}$	25865
$x_{22} : v_{25}-v_{23}$	818
$x_{23} : v_{25}-v_{20}$	19264
$x_{24} : v_{25}-v_{26}$	22877
$x_{25} : v_{25}-v_{24}$	24329
$x_{26} : v_{24}-v_{26}$	14073
$x_{27} : v_{24}-v_{19}$	16561
$x_{28} : v_{24}-v_6$	23114
$x_{29} : v_{20}-v_{26}$	17385
$x_{30} : v_{20}-v_{22}$	5676
$x_{31} : v_{20}-v_{18}$	19596
$x_{32} : v_{18}-v_{26}$	15523
$x_{33} : v_{18}-v_{19}$	16804
$x_{34} : v_{18}-v_{21}$	5677
$x_{35} : v_{21}-v_{19}$	13233
$x_{36} : v_{21}-v_{26}$	17397
$x_{37} : v_{19}-v_{26}$	15578
$x_{38} : v_1-v_{19}$	39855
$x_{39} : v_{13}-v_{19}$	40479

Hasil penelitian berupa jarak antar tempat wisata dapat dilihat pada Tabel 2, sedangkan graf yang terbentuk dari hasil penelitian dapat dilihat pada Gambar 1.

Tabel 2. Data Penelitian

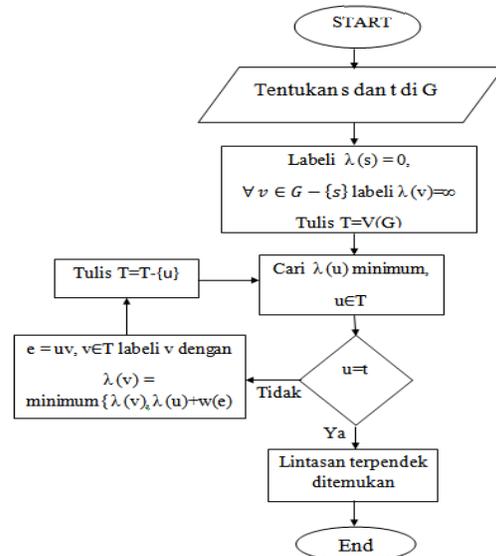
Sisi	Jarak (m)
$x_1 : v_1-v_{13}$	19511
$x_2 : v_{13}-v_{16}$	1410
$x_3 : v_8-v_{16}$	510
$x_4 : v_{13}-v_{17}$	3181
$x_5 : v_{17}-v_7$	3521
$x_6 : v_7-v_{10}$	7163
$x_7 : v_{10}-v_{12}$	14847
$x_8 : v_{12}-v_{11}$	4786
$x_9 : v_{11}-v_6$	683
$x_{10} : v_6-v_5$	1914
$x_{11} : v_5-v_{15}$	4351
$x_{12} : v_{15}-v_4$	2390
$x_{13} : v_4-v_3$	3358
$x_{14} : v_3-v_2$	1190



Gambar 1. Graf jaringan tempat wisata Kabupaten Gunungkidul

Pencarian rute terpendek menggunakan algoritma Dijkstra dapat digambarkan menggunakan *flowchart* yang dibuat bertujuan untuk mempermudah pembuatan *source code* program simulasi algoritma Dijkstra dalam menangani masalah rute terpendek pada graf yang dibangun menggunakan bahasa pemrograman Visual Basic 6.0. *Flowchart* juga akan membuat logika berfikir pembuatan *source code* program simulasi ini tersusun dengan baik berdasarkan urutan (step) yang sesuai dengan algoritma Dijkstra yang dipakai. *Flowchart* algoritma Dijkstra dapat dilihat pada Gambar 2.

Berdasarkan algoritma Dijkstra, maka untuk menentukan rute terpendek dari v1 (Pantai Sadeng) ke tempat wisata lainnya di Kabupaten Gunungkidul dengan menggunakan algoritma Dijkstra dapat dilihat pada Tabel 3.



Gambar 2. *Flowchart* algoritma Dijkstra

Tabel 3. Himpunan titik yang belum dilabel permanen

Titik (V _i)	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
λ(V _i)	0	∞	∞	∞	∞	∞	∞	∞	∞
T	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉

Titik (V _i)	V ₁₀	V ₁₁	V ₁₂	V ₁₃	V ₁₄	V ₁₅	V ₁₆	V ₁₇
λ(V _i)	∞	∞	∞	∞	∞	∞	∞	∞
T	V ₁₀	V ₁₁	V ₁₂	V ₁₃	V ₁₄	V ₁₅	V ₁₆	V ₁₇

Titik (V _i)	V ₁₈	V ₁₉	V ₂₀	V ₂₁	V ₂₂	V ₂₃	V ₂₄	V ₂₅
λ(V _i)	∞	∞	∞	∞	∞	∞	∞	∞
T	V ₁₈	V ₁₉	V ₂₀	V ₂₁	V ₂₂	V ₂₃	V ₂₄	V ₂₅

Titik (V _i)	V ₂₆	V ₂₇
λ(V _i)	∞	∞
T	V ₂₆	V ₂₇

$$(v_{13}) = \min \{ \infty, 0 + 19511 \} = \min \{ \infty, 19511 \} = 19511.$$

$$(v_{19}) = \min \{ \infty, 0 + 39855 \} = \min \{ \infty, 39855 \} = 39855.$$

Selanjutnya diiterasikan sampai titik terakhir yaitu tempat wisata terjauh dari Pantai Sadeng (v₁), sehingga diperoleh hasil iterasi ke-27 dapat dilihat pada Tabel 4.

Tabel 4. Himpunan semua titik yang dilabel permanen

Titik (V _i)	V ₁	V ₂	V ₃	V ₄	V ₅
λ(V _i)	0	66895	65705	62347	55606
T	-	-	-	-	-

Titik (V _i)	V ₆	V ₇	V ₈	V ₉	V ₁₀
λ(V _i)	53692	26213	21431	78303	33376
T	-	-	-	-	-

Titik (V _i)	V ₁₁	V ₁₂	V ₁₃	V ₁₄	V ₁₅
λ(V _i)	53009	48223	19511	77069	59957
T	-	-	-	-	-

Titik (V _i)	V ₁₆	V ₁₇	V ₁₈	V ₁₉	V ₂₀
λ(V _i)	20921	22692	56659	39855	72818
T	-	-	-	-	-

Titik (V _i)	V ₂₁	V ₂₂	V ₂₃	V ₂₄	V ₂₅
λ(V _i)	53088	78494	79128	56416	78310
T	-	-	-	-	-

Titik (V _i)	V ₂₆	V ₂₇
λ(V _i)	55433	74033
T	-	-

Dari tabel 4 dapat dilihat bahwa setiap titik di G sudah dilabel permanen. Tempat wisata terjauh dari Pantai Sadeng (v₁) adalah Air Terjun Srigethuk (v₂₃). Karena label permanen dari v₂₃ adalah λ (v₁, v₂₃) = 79128, jarak rute terpendek dari v₁ ke v₂₃ di graf bobot G adalah 79128 meter. Untuk menentukan rute terpendek dari Pantai Sadeng (v₁) ke Air Terjun Srigethuk (v₂₃) dengan melalui beberapa tempat wisata lain di Kabupaten Gunungkidul dapat dilakukan dengan metode telusur balik, yaitu dari (v₂₃) ke (v₁).

$$\lambda (v_1, v_{23}) = 79128 = 78310 + 818 = \lambda (v_1, v_{25}) + W (v_{25} v_{23})$$

$$\lambda (v_1, v_{25}) = 78310 = 55433 + 22877 = \lambda (v_1, v_{26}) + W (v_{26} v_{25})$$

$$\lambda (v_1, v_{26}) = 55433 = 39855 + 15578 = \lambda (v_1, v_{19}) + W (v_{19} v_{26})$$

$$\lambda (v_1, v_{19}) = 39855 = 0 + 39855 = \lambda (v_1) + W (v_1 v_{19})$$

Jadi,

$$\lambda (v_1, v_{23}) = W (v_1 v_{19}) + W (v_{19} v_{26}) + W (v_{26} v_{25}) + W (v_{25} v_{23})$$

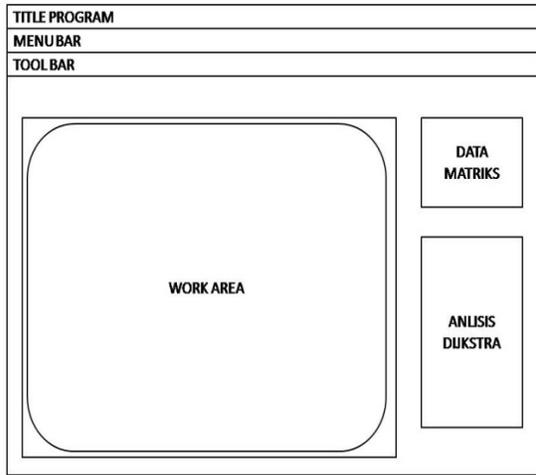
Sehingga diperoleh sebuah rute terpendek dari Pantai Sadeng (v₁) ke tempat wisata terjauh yaitu Air Terjun Srigethuk (v₂₃) di graf bobot G dengan rute v₁, v₁₉, v₂₆, v₂₅, v₂₃ dan jarak 79128 meter. Jadi dari Pantai Sadeng menuju Air Terjun Srigethuk dapat melalui beberapa tempat wisata lain diantaranya Desa Wisata Umbulrejo (v₁₉), Goa Pindul (v₂₆), Goa Rancang Kencono (v₂₆).

Dengan merubah titik awal dengan titik yang berbeda dan melakukan algoritma Dijkstra pada titik-titik yang lain, didapat rute terpendek tiap tiap objek wisata.

Program Simulasi

Desain tampilan dibuat untuk memudahkan programmer dalam menterjemahkan ke dalam bentuk bahasa pemrograman. Desain tampilan juga digunakan sebagai bahan acuan ketika membuat tampilan program simulasi algoritma Dijkstra dalam

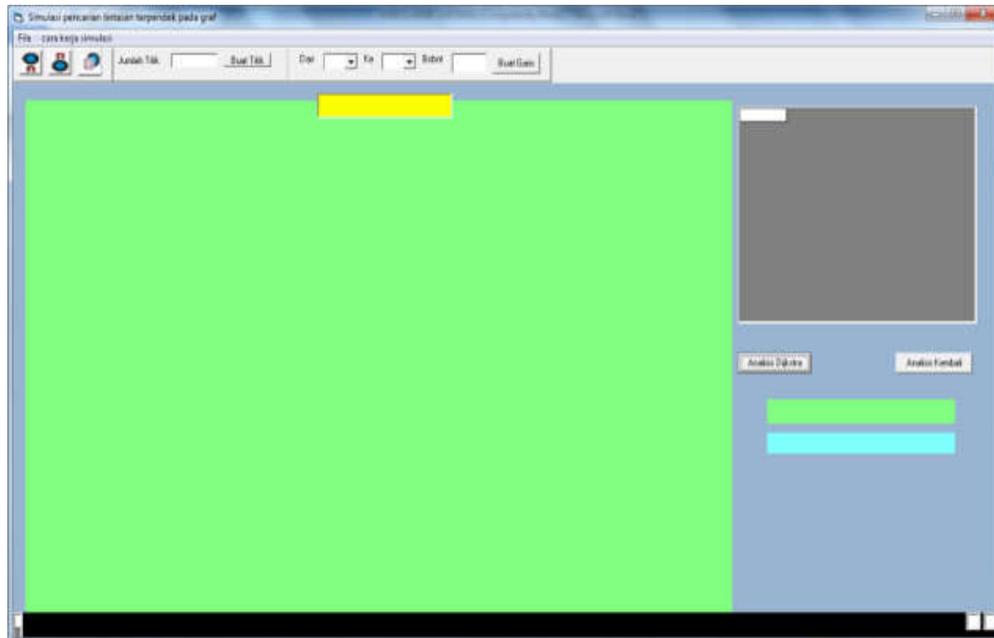
menangani masalah pencarian rute terpendek pada graf terstruktur dengan baik. Desain tampilan program dapat dilihat pada Gambar 3.



Gambar 3. Desain tampilan simulasi.

setiap titik pada *work area*. Selanjutnya adalah memberi nama titik tersebut dan memberi bobot pada rute antara titik yang satu dengan yang lainnya.

Data graf yang telah diinput pada form graf selanjutnya diproses untuk mendapatkan matriks jarak dari graf tersebut. Bobot jarak diproses lebih lanjut dengan tool analisis Dijkstra. Tool analisis Dijkstra dilengkapi *source code* yang menjadikan *tool* ini dapat menentukan rute terpendek menggunakan algoritma Dijkstra. Source code dari algoritma dijkstra dapat dilihat pada Gambar 5.



Gambar 4. Simulasi pencarian rute terpendek

Proses *input* graf dilakukan dengan cara menggambar titik dan rute yang menghubungkan

```
Private Sub cmdAnalisisDijkstra_Click()
    prepareFSP
    awl = getIndexofTabName(Awal)
```

```

    ahr = getIndexOfTabName(Akhir)
    If (awl = -1) Or (ahr = -1) Then
        MsgBox "Ada yang salah silahkan
cek kembali"
        Exit Sub
    End If
    flxS.Row = 1
    flxDist.Row = 1
    flxLintasan.Row = 1
    Dim MAX As Integer
    MAX = flxgraf.Cols
    Dim alur As Integer
    Dim jrk As Integer
    Dim i As Integer
    Dim min As Integer
    Dim pencarian As Boolean
    pencarian = True
    alur = awl
    jrk = 0
    flxS.TextMatrix(1, alur) = "True"
    flxS.Row = 1
    flxS.Col = alur
    flxS.CellForeColor = vbRed
    flxS.CellFontBold = True
    flxDist.TextMatrix(1, alur) = 0
    Do While pencarian
        If (min = INF) Then
            pencarian = False
        End If
        flxS.TextMatrix(1, alur) = "True"
        flxS.Row = 1
        flxS.Col = alur
        flxS.CellForeColor = vbRed
        flxS.CellFontBold = True
        For i = 1 To MAX - 1
            If (myVl(flxdist.TextMatrix(alur, i)) <>
0)And_
(myVl(flxdist.TextMatrix(1, i)) >
myVl(flxdist.TextMatrix(alur, i)) + jrk)
Then
            flxDist.TextMatrix(1, i) =
myVl(flxdist.TextMatrix(alur, i)) + jrk
            flxLintasan.TextMatrix(1, i) =
alur
        End If
        Next i
        min = INF
        For i = 1 To MAX - 1
            If (myVl(flxdist.TextMatrix(1,
i)) < min) And
(flxs.TextMatrix(1, i) = "False"))
Then
                min = myVl(flxdist.TextMatrix(1,
i))
                alur = i
                jrk = myVl(flxdist.TextMatrix(1,
i))
            End If
        Next i
    Loop
iRES_SIZE = 0
    makeAllLines_Black
    lblhasil.Caption = "Lintasan:"
    alur = ahr
    Do While alur <> awl
        If (flxLintasan.TextMatrix(1,
alur) = "0")Then
            lblhasil.Caption = "Tidak Ada
Lintasan " &
            flxgraf.TextMatrix(0, awl) & "
ke " &

```

```

flxgraf.TextMatrix(0, ahr) & "!"
        lbljarak.Caption = ""
        Exit Sub
    End If
    lblhasil.Caption =
    flxgraf.TextMatrix(0, alur)
    addTO_RESULT (alur)
    lblhasil.Caption =
    lblhasil.Caption & " <- "
    alur =
    myVl(flxLintasan.TextMatrix(1, alur))
    Loop
    lblhasil.Caption = lblhasil.Caption &
    flxgraf.TextMatrix(0, awl)
    addTO_RESULT (awl)
    lbljarak.Caption =
    flxDist.TextMatrix(1, ahr)
    markLINES
End Sub

```

Gambar 5. Source code algoritma Dijkstra

PENUTUP

Algoritma Dijkstra dapat digunakan dalam pencarian rute terpendek tempat wisata di Kabupaten Gunungkidul, dengan (v_i, v_j) adalah jarak terpendek dari v_i ke v_j . Contoh, Rute terpendek dari Pantai Sadeng (v_1) menuju titik terjauh dari Pantai Sadeng, yaitu Air Terjun Srigethuk (v_{23}) adalah sebagai berikut.

$$\lambda(v_1, v_{23}) = 79128 = 78310 + 818 = \lambda(v_1, v_{25}) + W(v_{25} v_{23})$$

$$\lambda(v_1, v_{25}) = 78310 = 55433 + 22877 = \lambda(v_1, v_{26}) + W(v_{26} v_{25})$$

$$\lambda(v_1, v_{26}) = 55433 = 39855 + 15578 = \lambda(v_1, v_{19}) + W(v_{19} v_{26})$$

$$\lambda(v_1, v_{19}) = 39855 = 0 + 39855 = (v_1) + W(v_1 v_{19})$$

Diperoleh,

$$\lambda(v_1, v_{23}) = W(v_1 v_{19}) + W(v_{19} v_{26}) + W(v_{26} v_{25}) + W(v_{25} v_{23})$$

Jadi, jarak rute terpendek dari titik V_1 (Pantai Sadeng) ke titik V_{23} (Air Terjun Srigethuk) adalah 79128 meter. Dengan sebuah rute terpendeknya adalah melalui titik-titik v_1 (Pantai Sadeng), v_{19} (Desa Wisata Umbulrejo), v_{26} (Goa Pindul), v_{25} (Goa Rancang Kencono), v_{23} (Air Terjun Srigethuk).

Simulasi algoritma Dijkstra untuk menangani masalah pencarian rute terpendek tempat wisata di Kabupaten Gunungkidul yang dibangun menggunakan bahasa pemrograman Visual Basic telah selesai dirancang dan direalisasikan. Simulasi tersebut mampu menemukan rute terpendek dan jarak minimum dari titik awal ke titik tujuan pada graf yang direpresentasikan ke dalam program simulasi.

Metode dalam mencari rute terpendek pada simulasi ini baru menggunakan satu

algoritma, selanjutnya dapat ditambah algoritma lain untuk mengetahui perbedaan jarak dan rute yang dihasilkan oleh masing-masing algoritma. Visual Basic memiliki batasan integer 32767, karena itu simulasi ini juga memiliki batasan integer 32767 sehingga diperlukan adanya program yang diharapkan mampu melebihi batasan integer tersebut. Simulasi yang dibangun masih berbentuk manual dan dijalankan secara *offline*, sehingga diperlukan adanya penelitian lain yang diharapkan mampu membangun simulasi secara *online*.

DAFTAR PUSTAKA

- Budayasa, I K. 2007. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Dewi, L.J.E. 2010. Pencarian Rute Terpendek Tempat Wisata di Bali dengan Menggunakan Algoritma Dijkstra. *Seminar Nasional Aplikasi Teknologi Informasi 2010* (ISSN: 1907-5022).
- Diana, O.P. dkk. 2011. Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A*) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto. *TRANSMISI*, Vol. 13, No. 1 (ISSN: 1411-0814). Tersedia di <http://ejournal.undip.ac.id/index.php/transmisi>.
- Dimiyati, T.T. & Dimiyati, A. 1999. *Operations Research Model-Model Pengambilan Keputusan*. Bandung: Sinar Baru Algensindo.
- Hillier S.F dan Lieberman J.G. 1990. *Pengantar Riset Operasi*. Jakarta: Erlangga.
- Kartika, G.Y & Jeffrey, T. 2002. Perencanaan Rute Perjalanan di Jawa Timur dengan Dukungan GIS Menggunakan Metode Dijkstra's. *Jurnal Informatika*, Vol. 3, No. 2, Nopember 2002, 68-73.
- Mulyono, S. 2004. *Riset Operasi*. Jakarta: Lembaga Penerbit Fakultas Ekonomi UI.
- Pradana, B. 2006. *Studi dan Implementasi Persoalan Lintasan Terpendek Suatu Graf dengan Algoritma Dijkstra dan Algoritma Bellman-ford*. Bandung: Institut Teknologi Bandung.
- Rahadian, F. 2011. Sistem Pengelolaan Database Siswa Menggunakan Pemrograman Visual Studio.net. *Jurnal Pendidikan Dompot Dhuafa edisi I/2011*.
- Ricky, D.A. dkk. 2013. Pembuatan Aplikasi Prakiraan Cuaca Menggunakan Bahasa Pemrograman Visual Basic. *Jurnal MIPA UNSRAT Online 2* (2) 128-132. Tersedia di <http://ejournal.unsrat.ac.id/index.php/jmuo>.
- Setyadi, A.H. 2011. *Dasar Pemrograman Visual Basic*. Portal Edukasi Indonesia Open Knowledge and Education. Tersedia di <http://shirotholmustaqim.files.wordpress.com/2010/02/dasar-pemrograman-visual-basic1.pdf>
- Sulindawati dkk. 2015. Pendistribusian Barang Farmasi Menggunakan Algoritma Dijkstra (Studi Kasus : PT. Air Mas Chemical). *Jurnal SAINTIKOM, Vol. 14, No. 1* (ISSN: 1978-6603), Januari 2015.
- Suparno. 2011. *Modul Pemrograman Visual Basic*. Palangkaraya: Sekolah Tinggi Manajemen Informatika dan Komputer Palangkaraya. Tersedia di <https://www.google.com/#q=Modul+pemrograman+visual+basic>
- Suyitno, H. 1997. *Pengantar Program Linear*. Semarang: FPMIPA IKIP Semarang.
- Taha, A.H. 1997. *Riset Operasi*. Jakarta: Bina Rupa Aksara.