Jurnal Kajian dan Terapan Matematika
Volume 8, Edisi 3, Bulan November 2022, 172-187
http://journal.student.uny.ac.id/ojs/index.php/jktm

Penentuan jalur kritis pada penjadwalan proyek konstruksi menggunakan fuzzy trapezoidal critical path method

Determination of the critical path in construction project scheduling using fuzzy trapezoidal critical path method

Saurika Muhsinina, Prodi Matematika FMIPA UNY
Sahid *, Prodi Matematika FMIPA UNY
*e-mail: sahid@uny.ac.id

Abstrak
Penelitian ini bertujuan untuk menjadwalkan proyek konstruksi dengan fuzzy trapezoidal critical path method (FCPM), mengetahui perilaku algoritma FCPM dalam menyelesaikan masalah penjadwalan proyek konstruksi, dan mengetahui hasil penyelesaian masalah penjadwalan proyek konstruksi dengan FCPM. Penelitian ini merupakan penelitian masalah penjadwalan dengan FCPM menggunakan program Python. Data penelitian diperoleh dengan teknik acak menggunakan sejumlah 12 aktivitas dan studi kasus proyek peningkatan jalan dan jembatan yang masing-masing memiliki 7 aktivitas. Data random diacak menggunakan Microsoft Excel dan data studi kasus diperoleh dari instansi pemerintah di bidang infrastruktur pembangunan di Sleman. Hasil penelitian menunjukkan bahwa 6 dari 12 aktivitas yaitu A-B-D-F-G-L merupakan jalur kritis pada data acak. Pada proyek peningkatan jalan, aktivitas mobilisasi, pekerjaan tanah, pekerjaan aspal, pekerjaan trotoar, dan pekerjaan lain-lain pada proyek peningkatan jalan merupakan aktivitas kritis. Durasi normal penyelesaian proyek jalan adalah 112 sampai 119 hari, dengan durasi tercepat 105 hari, dan terlama 126 hari. Hasil tersebut mendekati penyelesaian secara nyata di lapangan yang berlangsung selama 120 hari. Pada proyek peningkatan jembatan, aktivitas mobilisasi, struktur, dan pekerjaan lain-lain merupakan aktivitas kritis. Durasi normal penyelesaian proyek adalah 153 sampai 159 hari, dengan durasi tercepat 147 hari, dan terlama 166 hari. Hasil tersebut mendekati rencana penjadwalan yang akan dilaksanakan selama 150 hari.
Kata kunci: Penjadwalan, Fuzzy Critical Path Method (FCPM)

Abstract
The research aims to schedule construction projects using the fuzzy trapezoidal critical path method (FCPM), determine the behavior of the FCPM algorithm in solving construction project scheduling problems, and find out the results of solving construction project scheduling problems using the FCPM. This research is a scheduling problem research with FCPM using the Python program. The research data was obtained by random technique using 12 activities and case studies of road and bridge improvement projects, each of which has 7 activities. Random data is randomized using Microsoft Excel and case study data obtained from government agencies in the infrastructure development department in Sleman. The results showed that 6 of the 12 activities, namely A-B-D-F-G-L, were critical paths in random data. In the road improvement project, the mobilization, earthwork, asphalt work, sidewalk work, and other works on the road improvement project are all critical activities. The normal duration of the project completion is 112 to 119 days with the fastest and the longest duration are 105 and 126 days respectively. These results are close to the actual completion in the project which lasts for 120 days. In the bridge improvement project, the mobilization, the structure work, and other works are critcial activities. The normal duration of project completion is 153 to 159 days with the fastest and the longest duration are 147 and 166 days respectively. The results are close to the scheduling plan that will be implemented for 150 days.
Keywords*: Scheduling, Fuzzy Critical Path Method (FCPM)*

PENDAHULUAN

Proyek konstruksi adalah realisasi perencanaan dari hubungan antara *partner*, sarana teknis dan material dengan tujuan untuk mewujudkan suatu infrastruktur (Nový, Nováková, & Waldhans, 2012). Secara umum terdapat dua macam konstruksi yaitu konstruksi bangunan gedung (bangunan perumahan, gedung, hotel) dan konstruksi bangunan sipil (jembatan, jalan, lapangan terbang, terowongan, irigasi, bendungan) (Rani, 2016).

Pembangunan suatu proyek konstruksi memerlukan perencanaan yang dilakukan secara menyeluruh agar berjalan dengan baik. Tahap perencanaan proyek konstruksi yang dimaksud berupa perencanaan gambar (*master plan*), perencanaan material (bahan bangunan), perencanaan jumlah dan keahlian tenaga kerja, perencanaan pembiayaan, serta perencanaan waktu pelaksanaan. Pada perencanaan tersebut, terdapat rincian detail aktivitas-aktivitas yang dilakukan dan harus sesuai dengan desain *master plan* yang dirancang di awal proyek. Beberapa aktivitas dalam proyek dapat dikerjakan secara bersamaan (paralel) dengan kegiatan lainnya, namun terdapat aktivitas yang dapat dikerjakan jika kegiatan lain sudah diselesaikan (seri), dan terdapat juga aktivitas yang dapat dikerjakan kapan saja tanpa terpengaruh kegiatan lain. Adanya berbagai macam aktivitas proyek dan durasi yang berbeda-beda menyebabkan perlunya penjadwalan proyek. Penjadwalan proyek adalah alat untuk menentukan kegiatan yang diperlukan dalam menyelesaikan suatu proyek dengan urutan dan kerangka waktu tertentu, dimana setiap aktivitas harus dilaksanakan agar proyek selesai tepat waktu dengan biaya yang ekonomis (Atin & Lubis, 2019).

Agar tercapai penjadwalan yang terstruktur, maka perlu adanya perencanaan jaringan kerja (*network planning*). Jaringan proyek adalah alat yang mudah dan dapat diakses untuk penghitungan lintasan kritis guna menentukan waktu mulai aktivitas paling awal dan terakhir dari fase penjadwalan (Vanhoucke, 2012). Jaringan proyek terfokus pada rincian detail aktivitas, kegiatan pendahulunya dan durasinya. Tujuan dibentuk jaringan proyek yaitu untuk mengetahui umur proyek dengan menentukan aktivitas-aktivitas kritisnya. Aktivitas kritis adalah aktivitas yang apabila terjadi penundaan akan mempengaruhi waktu penyelesaian keseluruhan proyek. Aktivitas kritis menjadi perhatian, sebab terlambat atau tidaknya suatu proyek tergantung pada jumlah lintasan kritis yang berpengaruh pada umur proyek. Salah satu algoritma atau metode penyelesaian yang biasa digunakan untuk menyelesaikan masalah tersebut adalah algoritma jalur kritis atau *Critical Path Method* (CPM). CPM merupakan metode berbasis jaringan yang dirancang untuk membantu dalam perencanaan, penjadwalan, dan pengendalian proyek (Taha, 2017). CPM bertujuan untuk mengidentifikasi aktivitas kritis di jalur kritis sehingga sumber daya mungkin dikonsentrasikan pada aktivitas ini untuk mengurangi waktu lama proyek (Nowpada, Rao, & Veeramachaneni, 2010) . Kendala yang terjadi pada saat di lapangan CPM kadang tidak berjalan sesuai rencana karena beberapa faktor insidental, seperti cuaca ekstrim, kekurangan bahan konstruksi, perubahan material, kerusakan peralatan, keterlambatan pengiriman bahan, dan lain-lain, sehingga terdapat aktivitas yang berubah durasinya.

Kekurangan metode CPM tidak dapat mengatasi aktivitas proyek yang sewaktu-waktu berubah durasinya. Apabila masalah tersebut tidak diselesaikan maka proyek akan berjalan tidak sesuai rencana, menyebabkan penambahan biaya, dan memperlambat proyek. Sebagai alternatif untuk mengatasi masalah ketidakpastian perubahan durasi aktivitas tersebut, dapat digunakan metode *Fuzzy Critical Path Method* (FCPM). FCPM merupakan metode untuk mengidentifikasi lintasan kritis pada jaringan proyek dengan durasi aktivitas yang belum pasti menggunakan bilangan *fuzzy* atau interval *fuzzy* (Nasution, 1996). FCPM dapat menjadwalkan proyek dengan meminimalkan umur proyek (durasi tercepat proyek), memenuhi standar durasi normal, dan batas maksimal umur proyek. Dalam penelitian ini masalah penjadwalan difokuskan pada penyelesaian FCPM menggunakan bilangan *trapezoidal fuzzy* dengan

mempertimbangkan empat estimasi waktu, yakni waktu tercepat (optimis), rentang durasi normal (paling mungkin), dan terlama (pesimis).

Algoritma penjadwalan menggunakan FCPM dengan tujuan untuk mengidentifikasi aktivitas dalam proyek adalah menetapkan hubungan prioritas (aktivitas pendahulu) dari semua aktivitas, memperkirakan waktu *fuzzy* setiap aktivitas, membuat jaringan kerja, menghitung *earliest time* dan *latest time* untuk mengetahui nilai *slack time* atau waktu longgar. Aktivitas yang tidak memiliki waktu longgar merupakan aktivitas kritis yang perlu diperhatikan agar tidak terjadi penundaan sehingga proyek selesai tepat waktu. Algoritma FCPM dapat dituliskan dan diselesaikan dengan pemrograman komputer.

Bahasa pemrograman diperlukan untuk mempermudah dalam menjalankan perintah serangkaian kode program tertentu. Bahasa pemrograman membantu pekerjaan yang memerlukan sistem agar bekerja secara otomatis. Terdapat berbagai jenis bahasa pemrograman, seperti: JavaScript, PowerShell, C++, Java, PHP, Python, Go, T-SQL, dan C# (Johnson, 2019). Pada penelitian ini akan digunakan bahasa pemrograman Python untuk menyelesaikan algoritma FCPM. Python adalah bahasa fungsional yang kuat, prosedural, berorientasi objek, yang dibuat di akhir 1980-an oleh Guido Van Rossum dan bersifat *open source* (Bhasin, 2018). Python sangat menekankan pada keterbacaan kode dan kesederhanaan, sehingga memungkinkan *programmer* untuk mengembangkan aplikasi dengan cepat (Chan, 2014). Bahasa Python juga mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya.

METODE

Langkah-langkah Penelitian

Langkah-langkah penelitian dalam menentukan jalur kritis *fuzzy* pada penjadwalan proyek konstruksi adalah merumuskan permasalahan penjadwalan proyek secara matematis, merancang algoritma untuk menyelesaikan masalah penentuan jalur kritis, menyusun program komputer Python berdasarkan algoritma yang dikembangkan, menguji coba program menggunakan data acak dan data studi kasus, serta menganalisis hasil uji coba program.

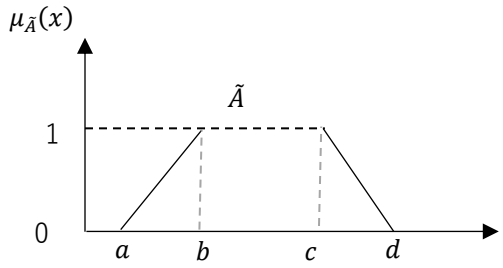
Pemodelan Masalah secara Matematis

Bilangan *fuzzy* $\tilde{A}$ adalah himpunan *fuzzy* dengan fungsi keanggotaan $\mu_{\tilde{A}}(x)$ yang *piecewise continuous*, konvek, dan normal, paling tidak salah satu elemen x derajat keanggotaannya adalah 1 (Nowpada, Rao, & Veeramachaneni, 2010).

Definisi: Bilangan *fuzzy* dengan fungsi keanggotaan sebagai berikut:

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x-a}{b-a}, & \text{untuk } a \le x \le b \\ 1, & \text{untuk } b \le x \le c \\ \dfrac{d-x}{d-c}, & \text{untuk } c \le x \le d \\ 0, & \text{untuk lainnya} \end{cases} \quad (1)$$

disebut bilangan *trapezoidal fuzzy* $\tilde{A} = (a, b, c, d)$. Kurva fungsi keanggotaan *trapezoidal fuzzy* dapat dilihat pada gambar 1.



Gambar 1. Kurva bilangan *fuzzy* trapesium

Misalkan $\widetilde{B_1}$ dan $\widetilde{B_2}$ adalah dua bilangan *fuzzy* trapesium dengan $\widetilde{B_1} = (a_1, b_1, c_1, d_1)$ dan $\widetilde{B_2} = (a_2, b_2, c_2, d_2)$, maka operasi bilangan *trapezoidal fuzzy* adalah sebagai berikut (Nowpada, Veeramachaneni, & Rao, 2010).

$$\widetilde{B_1} \oplus \widetilde{B_2} = (a_1, b_1, c_1, d_1) \ominus (a_2, b_2, c_2, d_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2) \quad (2)$$

$$\widetilde{B_1} \ominus \widetilde{B_2} = (a_1, b_1, c_1, d_1) \ominus (a_2, b_2, c_2, d_2) = (a_1 - d_2, b_1 - c_2, c_1 - b_2, d_1 - a_2) \quad (3)$$

Pengidentifikasian masalah penentuan jalur kritis pada penjadwalan proyek secara matematis dirumuskan sebagai berikut.

$$a_i \le b_i \le c_i \le d_i \text{ satuan waktu.} \quad (4)$$

$$\tilde{E}^s_{(1)} = \widetilde{TE}_{(1)} = (0,0,0,0). \quad (5)$$

$$\tilde{E}^f_{(i,j)} = \tilde{E}^s_{(i)} \oplus \tilde{t}_{(i,j)}. \quad (6)$$

$$\widetilde{TE}_{(A_j)} = \max\left(\left(\tilde{E}^s_{(A_{i_1})} \oplus \tilde{t}_{(A_{i_1,j})}\right), \left(\tilde{E}^s_{(A_{i_2})} \oplus \tilde{t}_{(A_{i_2,j})}\right), \dots, \left(\tilde{E}^s_{(A_{i_n})} \oplus \tilde{t}_{(A_{i_n,j})}\right)\right) \quad (7)$$

$$\widetilde{TE}_{(A_n)} = \widetilde{TL}_{(A_n)} = \tilde{L}^f_{(A_j,A_n)}. \quad (8)$$

$$\tilde{L}^s_{(i,j)} = \tilde{L}^f_{(j)} \ominus \tilde{t}_{(i,j)}. \quad (9)$$

$$\widetilde{TL}_{(A_i)} = \min\left(\left(\tilde{L}^f_{(A_{j_1})} \ominus \tilde{t}_{(A_{i,j_1})}\right), \left(\tilde{L}^f_{(A_{j_2})} \ominus \tilde{t}_{(A_{i,j_2})}\right), \dots, \left(\tilde{L}^f_{(A_{j_n})} \ominus \tilde{t}_{(A_{i,j_n})}\right)\right) \quad (10)$$

$$\widetilde{TL}_{A_1} = (a_1, b_1, c_1, d_1) = (0,0,0,0). \quad (11)$$

$$\widetilde{TS}_{ij} = \widetilde{TL}_j \ominus \tilde{t}_{ij} \ominus \widetilde{TE}_i. \quad (12)$$

$$Mag\left(\tilde{A}\right) = \frac{a + 5b + 5c + d}{12}. \quad (13)$$

Keterangan:

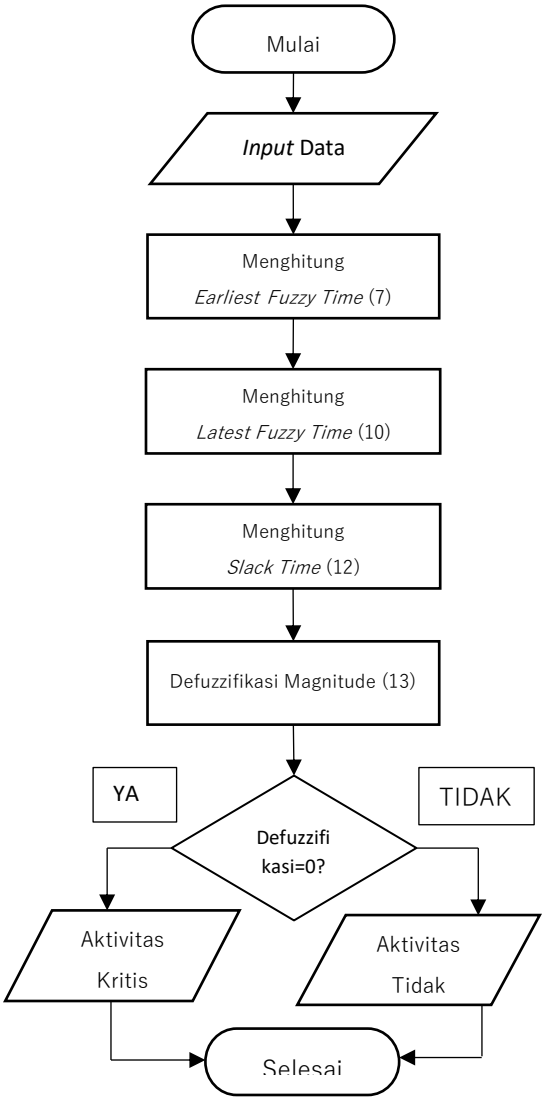
A : himpunan dari semua aktivitas atau *node* dalam jaringan proyek,
$\tilde{t}_i$: waktu aktivitas *fuzzy* dari A_i,
$\tilde{E}^s_{(i)}$: waktu *earliest start fuzzy* dari A_i,
$\tilde{E}^f_{(i)}$: waktu *earliest finish fuzzy* dari A_i,
$\tilde{L}^s_{(i)}$: waktu *latest start fuzzy* dari A_i,
$\tilde{L}^f_{(i)}$: waktu *latest finish fuzzy* dari A_i,
$\widetilde{TE}$: waktu total *earliest fuzzy*,
$\widetilde{TL}$: waktu total *latest fuzzy*,
$\widetilde{TS}_i$: waktu *slack fuzzy* dari A_i,
$Mag\ (\tilde{A}_i)$: defuzzifikasi dari A_i.

Perancangan Algoritma

Perancangan algoritma diperlukan untuk menyelesaikan masalah penentuan jalur kritis pada penjadwalan proyek. Algoritma untuk menentukan jalur kritis *fuzzy* adalah sebagai berikut.

1. Menghitung perhitungan maju menggunakan persamaan (7).
2. Menghitung perhitungan mundur menggunakan persamaan (10).
3. Menghitung kelonggaran waktu (*slack time*) menggunakan persamaan (12).
4. Menghitung defuzzifikasi menggunakan metode *magnitude* $Mag(\tilde{A})$ pada persamaan (13).
5. Menentukan aktivitas kritis ditandai dengan hasil defuzzifikasi $\tilde{A} = 0$.
6. Menemukan *fuzzy critical path* (jalur kritis *fuzzy*) dengan menggunakan aktivitas-aktivitas kritis yang saling terhubung.

Gambar 2 menunjukkan algoritma FCPM atau langkah-langkah dalam menentukan jalur kritis *fuzzy*.



Gambar 2. Rancangan Algoritma FCPM

Penyusunan Program Python

Penyusunan program berikut merupakan algoritma FCPM yang dikembangkan menggunakan bahasa Python dengan bantuan *software* Visual Studio Code. Program ini dibuat dengan beberapa ketentuan *input* data. Data yang akan dimasukkan terdiri dari tujuh kolom judul yaitu DESC, ACTIVITY, PREDECESSORS, F1, F2, F3, F4. DESC menunjukkan nama aktivitas (boleh dimasukkan boleh tidak). ACTIVITY mewakili nama aktivitas yang

disimbolkan dengan huruf kapital. PREDECESSORS menunjukkan aktivitas pendahulu yang juga disimbolkan dengan huruf kapital (harus terdapat pada kolom *activity*). F1, F2, F3, F4 menunjukkan durasi aktivitas dengan bilangan *fuzzy* trapesium. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka disini F1 bernilai a, F2 bernilai b, F3 bernilai c, dan F4 bernilai d. Berikut ini penjelasan mengenai penyusunan *script*.

1. Perhitungan *forward pass (earliest fuzzy time)*

Nilai pada (ES1, ES2, ES3, ES4) menunjukan nilai *Earliest Start* dalam bentuk bilangan *fuzzy* trapesium. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka ES1 bernilai a, ES2 bernilai b, ES3 bernilai c, dan ES4 bernilai d. Nilai pada (EF1, EF2, EF3, EF4) juga berbentuk bilangan *fuzzy* trapesium yang menunjukan nilai *Earliest Finish*. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka EF1 bernilai a, EF2 bernilai b, EF3 bernilai c, dan EF4 bernilai d. Nilai ES berikutnya mengambil nilai maksimum dari perhitungan.

```
def forwardpass(self, data):
    self.data = data
    ntask = data.shape[0]
    ES1=np.zeros(ntask,dtype=np.uint8)
    ...
    EF4=np.zeros(ntask,dtype=np.uint8)
    temp1 = []
    ...
    temp4 = []
    for i in range(ntask):
        if(data['PREDECESSORS'][i]== None):
            ES1[i] = 0
            ...
            ES4[i] = 0
            try:
                EF1[i] = ES1[i] + data['F1'][i]
                ...
                EF4[i] = ES4[i] + data['F4'][i]
            except:
                self.errorFuzzyMsg()
        else:
            for j in data['PREDECESSORS'][i]:
                index = self.getTaskActivity(data,j)
                temp1.append(EF1[index])
                ...
                temp4.append(EF4[index])
                if(index==i):
                    self.errorPredMsg()
                else:
                    temp1.append(EF1[index])
                    ...
                    temp4.append(EF4[index])
            ES1[i] = max(temp1)
            ...
            ES4[i] = max(temp4)
            try:
                EF1[i] = ES1[i] + data['F1'][i]
                ...
```

```
            EF4[i] = ES4[i] + data['F4'][i]
        except:
            self.errorFuzzyMsg()
        temp1 = []
        ...
        temp4 = []
    data['ES1'] = ES1
    ....
    data['EF4'] = EF4
    return data
```

2. Perhitungan *backward pass (latest fuzzy time)*

Nilai (LS1, LS2, LS3, LS4) menunjukan nilai *Latest Start* dalam bentuk bilangan *fuzzy* trapesium. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka disini LS1 bernilai a, LS2 bernilai b, LS3 bernilai c, dan LS4 bernilai d. Nilai (LF1, LF2, LF3, LF4) menunjukan nilai *Latest Finish* dalam bentuk bilangan *fuzzy* trapesium. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka disini LF1 bernilai a, LF2 bernilai b, LF3 bernilai c, dan LF4 bernilai d. Nilai LF berikutnya mengambil nilai minimum dari perhitungan..

```
def backwardpass(self, data):
    self.data = data
    ntask = data.shape[0]
    temp1 = []
    ...
    temp4 = []
    LS1 = np.zeros(ntask, dtype=np.int8)
    ...
    LF4 = np.zeros(ntask, dtype=np.int8)
    SUCCESSORS = np.empty(ntask, dtype=object)
    for i in range(ntask-1,-1,-1):
        if(data['PREDECESSORS'][i] != "Start"):
            for j in data['PREDECESSORS'][i]:
                index=self.getTaskActivity(data,j)
                if(SUCCESSORS[index] !=None):
                    SUCCESSORS[index] += data['ACTIVITY'][i]
                else:
                    SUCCESSORS[index] = data['ACTIVITY'][i]
    data['SUCCESSORS'] = SUCCESSORS
    for i in range(ntask-1,-1,-1):
        if(data['SUCCESSORS'][i] == None):
            LF1[i] = np.max(data['EF1'])
            ...
            LF4[i] = np.max(data['EF4'])
            try:
                LS1[i] = LF1[i] - data['F4'][i]
                ...
                LS4[i] = LF4[i] - data['F1'][i]
            except:
                self.errorFuzzyMsg()
        else:
            for j in data['SUCCESSORS'][i]:
                index = self.getTaskActivity(data,j)
```

```
            temp1.append(LS1[index])
            ...
            temp4.append(LS4[index])
            if(index==i):
               self.errorPredMsg()
            else:
               temp1.append(LS1[index])
               ...
               temp4.append(LS4[index])
         LF1[i] = min(temp1)
         ...
         LF4[i] = min(temp4)
         try:
            LS1[i] = LF1[i] - data['F4'][i]
            ...
            LS4[i] = LF4[i] - data['F1'][i]
         except:
            self.errorFuzzyMsg()
      temp1 = []
      ...
      temp4 = []
   data['LS1'] = LS1
   ...
   data['LF4'] = LF4
   return data
```

3. Perhitungan nilai *slack time*

Nilai (S1, S2, S3, S4) menunjukan nilai *slack time* dalam bentuk bilangan *fuzzy* trapesium. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka S1 bernilai a, S2 bernilai b, S3 bernilai c, dan S4 bernilai d. Berikut *function* perhitungan *slack time*.

```
def slack(self, data):
   self.data = data
   ntask = data.shape[0]
   S1=np.zeros(shape=ntask,dtype=np.int8)
   ...
   S4=np.zeros(shape=ntask,dtype=np.int8)
   for i in range(ntask):
      S1[i]=data['LS1'][i] - data['ES4'][i]
      ...
      S4[i]=data['LS4'][i] - data['ES1'][i]
   data['S1'] = S1
   ...
   data['S4'] = S4
   return data
```

4. Perhitungan defuzzifikasi

Proses defuzzufikasi menggunakan metode *magnitude* untuk menentukan nilai *crisp* dari bilangan *trapezoidal fuzzy*. Jika (a, b, c, d) merupakan bilangan *fuzzy* trapesium, maka S1 bernilai a, S2 bernilai b, S3 bernilai c, dan S4 bernilai d, sedangkan rumus *magnitude* seperti pada persamaan (14).

```
def defuzzification(self, data):
   self.data = data
```

```
   ntask = data.shape[0]
   DEFUZZIFICATION=np.zeros(shape=ntask,dtype=np.float16)
   CRITICAL=np.empty(shape=ntask,dtype=object)
   a=data['S1']
   b=data['S2']
   c=data['S3']
   d=data['S4']
   for i in range(ntask):
      DEFUZZIFICATION[i]=(a[i]+5*b[i]+5*c[i]+d[i])/12
      if (DEFUZZIFICATION[i]==0):
         CRITICAL[i]='YES'
      else:
         CRITICAL[i]='NO'
   data['DEFUZZIFICATION'] = DEFUZZIFICATION
   data['CRITICAL'] = CRITICAL
   data=data.reindex(columns=['DESCR','ACTIVITY','PREDECESSORS','SUCCESSORS','
DAYS','ES','EF','LS','LF','SLACK','DEFUZZIFICATION','CRITICAL'])
   return data
```

5. Visualisasi diagram jalur kritis

Visualisasi diagram jalur kritis pada jaringan proyek diperlukan *function* sebagai berikut.

```
def vis(self, data, ntask):
   self.data = data
   self.ntask = ntask
   cp = []
   for i in range(ntask):
      if(data['DEFUZZIFICATION'][i]==0):
         cp.append(data['ACTIVITY'][i])
   listActivity = data["ACTIVITY"].values.tolist()
   listPredecessors = data["PREDECESSORS"].values.tolist()
   G = nx.DiGraph()
   G.add_nodes_from(listActivity)
   for i in range(0,len(listPredecessors)):
      if (len(listPredecessors[i]) != 1 and listPredecessors[i] != "Start"):
         for x in listPredecessors[i]:
            G.add_edge(x,listActivity[i])
      else:
         G.add_edge(listPredecessors[i], listActivity[i])
   pos_nodes=nx.nx_agraph.graphviz_layout(G, prog='dot')
   fig, ax = plt.subplots(figsize=(10, 12))
   crit_edges=[(n,cp[i+1]) for i, n in enumerate(cp[:-1])]
   nx.draw_networkx_edges(G, edgelist=crit_edges, pos=pos_nodes, width=10, alpha=0.5,
edge_color='r')
   nx.draw(G, pos=pos_nodes, with_labels = True, arrowsize=20, node_size=1000,
node_color='lightgreen')
         plt.show()
```

Persiapan Data

Data yang perlu disiapkan untuk menentukan jalur kritis *fuzzy* adalah aktivitas-aktivitas proyek, aktivitas pendahulu (*predecessor*), dan durasi *fuzzy* trapesium setiap aktivitas. Aktivitas dalam proyek beserta atributnya dapat disusun seperti pada tabel 1. Kolom *activity* mewakili

deskripsi aktivitas dalam proyek. Aktivitas pendahulu merupakan aktivitas yang harus dilakukan sebelum aktivitas berikutnya dilaksanakan. Aktivitas pendahulu juga disimbolkan dengan huruf kapital (yang terdapat dalam *activity*). Data durasi dalam bentuk bilangan *crisp* perlu diubah menjadi bilangan *trapezoidal fuzzy* untuk mengestimasikan rentang durasi normal, durasi tercepat dan durasi terlambat.

Tabel 1. Data Aktivitas dalam Proyek

DESCR	ACTIVITY	PREDECESSORS	F1	F2	F3	F4
…	A_1	…	a_1	b_1	c_1	d_1
…	A_2	…	a_2	b_2	c_2	d_2
…	…	…	…	…	…	…
…	A_n	…	a_n	b_n	c_n	d_n

Teknik Pengumpulan Data

1. Data Acak

Penggunaan data acak dilakukan menggunakan formula random pada Microsoft Excel. Nama aktivitas pada kolom *activity* dapat diurutkan sesuai abjad menggunakan *formula* "character". Kolom pada *predecessors* dapat mengambil abjad yang berada pada daftar kolom di *activiy* menggunakan *formula* "index" dan "randbetween" . Nilai F1 – F4 dapat diacak sesuai rentang yang dinginkan menggunakan *formula* "randbetween". Pada nilai F1 – F4 akan menghasilkan nilai yang acak, sehingga saat akan diolah, diperlukan mengurutkan nilai dari yang terkecil terlebih dahulu membentuk $F1 < F2 < F3 < F4$. Cara tersebut dapat menggunakan fitur *sort & filters → sort options → smallest to largest → sort left to right*.

2. Data Studi Kasus

Data studi kasus Proyek Peningkatan Jalan Tahun 2021 dan Proyek Peningkatan Jembatan Tahun 2022 diperoleh dari instansi pemerintah di bidang infrastruktur pembangunan di Sleman.

Alat-alat Penelitian

Alat yang digunakan pada penelitian ini yaitu Microsoft Word 2016, Microsoft Excel 2016, *Visual Studio Code versi 1.67.2,* dan *Python versi 3.8.5 (64-bit)* dengan menggunakan beberapa *package* yaitu pandas, numpy, os, time, matplotlib.pyplot, dan networkx.

HASIL DAN PEMBAHASAN

Fokus penelitian ini untuk mengetahui bagaimana perilaku algoritma FCPM dalam menyelesaikan masalah penentuan jalur kritis pada penjadwalan proyek. Untuk mengetahui hal ini, diperlukan uji coba program. Program yang dikembangkan yaitu Python. Uji coba program dilakukan menggunakan data acak dan contoh kasus proyek peningkatan jalan dan jembatan.

Hasil

Data yang akan diolah pada pengujian program dipastikan sudah siap berupa *file* dengan ekstension ods (**.ods**) beserta letak *sheet*-nya. Berikut ini merupakan tampilan hasil *run script* Python yang telah disusun.

Ketik tempat file excel yang akan dieksekusi(Extension: .ods):… .ods
Ketik nama sheet yang mau di eksekusi: ...

Pada kesempatan uji coba program ini, digunakan data acak dan studi kasus sebagai *input*-an. Pengujian program FCPM menggunakan Python memberikan hasil sebagai berikut.

1. Data Acak (*Sample Test*)

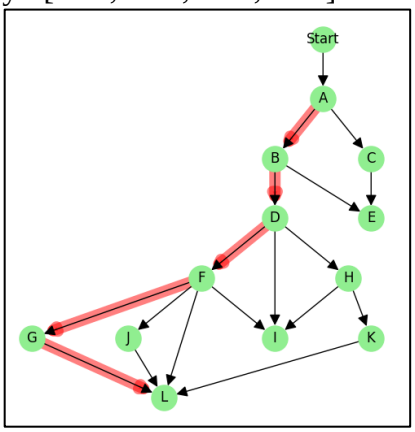
Data yang ditampilkan pada tabel uji coba merupakan data yang sudah diurutkan nilainya membentuk $F1 < F2 < F3 < F4$. Pada uji coba data acak ditentukan aktivitas sejumlah n=12 aktivitas dengan masing-masing memiliki maksimal 4 kegiatan pendahulu (*predecessors*). Setelah diacak menggunakan Microsoft Excel seperti pada metode penelitian, diperoleh aktivitas proyek, aktivitas pendahulu, dan durasi proyek seperti pada tabel 2.

Tabel 2. Data Aktivitas Acak

NO	ACTIVITY	PREDECESSORS	F1	F2	F3	F4
1	A	Start	3	5	5	6
2	B	A	4	4	5	5
3	C	A	5	5	6	6
4	D	B	5	6	6	8
5	E	BC	5	6	7	8
6	F	D	8	8	10	11
7	G	F	8	9	10	10
8	H	D	5	6	7	7
9	I	DHF	6	7	8	8
10	J	F	3	3	3	5
11	K	H	8	8	10	10
12	L	GFJK	8	9	9	9

Hasil yang diperoleh dari identifikasi jalur kritis pada tabel 2 adalah sebagai berikut.

```
TRAPEZOIDAL FUZZY CRITICAL PATH METHOD CALCULATOR
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
ES=Earliest Start; EF=Earliest Finish; LS=Latest Start, LF=Latest Finish
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    NO ACTIVITY PREDECESSORS   F1  ... S3  S4 DEFUZZIFICATION  CRITICAL
0   1.0        A       Start  3.0  ...  4  13       0.000000      YES
1   2.0        B           A  4.0  ...  4  13       0.000000      YES
2   3.0        C           A  5.0  ... 29  36      26.000000       NO
3   4.0        D           B  5.0  ...  4  13       0.000000      YES
4   5.0        E          BC  5.0  ... 29  36      26.000000       NO
5   6.0        F           D  8.0  ...  4  13       0.000000      YES
6   7.0        G           F  8.0  ...  4  13       0.000000      YES
7   8.0        H           D  5.0  ...  7  16       3.083984       NO
8   9.0        I         DHF  6.0  ... 15  23      10.914062       NO
9  10.0        J           F  3.0  ... 10  18       6.250000       NO
10 11.0        K           H  8.0  ...  7  16       3.083984       NO
11 12.0        L        GFJK  8.0  ...  4  13       0.000000      YES
[12 rows x 30 columns]
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Jalur Kritisnya adalah: A - B - D - F - G - L
Total durasi project dalam fuzzy:  [36.0, 41.0, 45.0, 49.0]  unit time
```



Gambar 3. Jalur Kritis Data Acak

Berdasarkan gambar 3 dan hasil yang diperoleh, menunjukkan bahwa A - B - D - F - G – L merupakan jalur kritis proyek dengan durasi normal 41-45 hari, tercepat 36 hari, dan terlambat 49 hari.

2. Data Studi Kasus

Proyek Peningkatan Jalan

Data yang diperoleh pada proyek peningkatan jalan yaitu seperti pada tabel 3.

Tabel 3. Data Aktivitas Proyek Jalan

DESCRIPTION	ACTIVITY	PREDECESSORS	F1	F2	F3	F4
Mobilisasi	A	Start	4	5	6	7
Drainase	B	A	36	38	40	42
Pekerjaan Tanah	C	A	43	45	47	49
Struktur	E	A	24	25	26	28
Pekerjaan Aspal	D	BCE	36	38	40	42
Pekerjaan Trotoar	F	D	18	19	20	21
Pekerjaan Lain-lain	G	F	4	5	6	7

Hasil yang diperoleh dari identifikasi jalur kritis pada tabel 3 adalah sebagai berikut.

TRAPEZOIDAL FUZZY CRITICAL PATH METHOD CALCULATOR

* *

ES=Earliest Start; EF=Earliest Finish; LS=Latest Start, LF=Latest Finish

* *

DESCRIPTION ACTIVITY PREDECESSORS F1 ... S4 DEFUZZIFICATI CRITICAL

Mobilisasi A Start 4.0 ... 21 0.0000 YES

Drainase B A 36.0 ... 28 7.0000 NO

Pekerjaan tanah C A 43.0 ... 21 0.0000 YES

Struktur E A 24.0 ... 40 20.4218 NO

Pekerjaan aspal D BCE 36.0 ... 21 0.0000 YES

Pekerjaan trotoar F D 18.0 ... 21 0.0000 YES

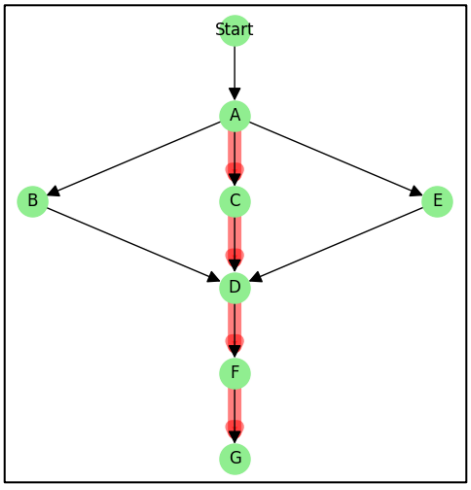
Pekerjaan lain G F 4.0 ... 21 0.0000 YES

[7 rows x 30 columns]

* *

Jalur Kritisnya adalah: A - C - D - F - G

Total durasi project dalam fuzzy: [105.0, 112.0, 119.0, 126.0] unit time



Gambar 4. Jalur Kritis Proyek Jalan

Berdasarkan gambar 4 dan hasil yang diperoleh, menunjukkan bahwa A-C-D-F-G merupakan jalur kritis proyek dengan durasi normal 112-119 hari, tercepat 105 hari, dan terlambat 126 hari.

Proyek Pembangunan Jembatan

Data yang diperoleh pada proyek peningkatan jembatan yaitu seperti pada tabel 4.

Tabel 4. Data Aktivitas Proyek Jembatan

DESCRIPTION	ACTIVITY	PREDECESSORS	F1	F2	F3	F4
Mobilisasi	A	Start	29	30	31	33
Pekerjaan Tanah	C	A	31	32	33	35
Struktur	F	A	100	104	108	112
Drainase	B	C	50	52	54	56
Perkerasan Berbutir	D	C	4	5	6	7
Perkerasan Aspal	E	D	4	5	6	7
Pekerjaan Lain-lain	G	EF	18	19	20	21

Hasil yang diperoleh dari identifikasi jalur kritis pada tabel 4 adalah sebagai berikut.

TRAPEZOIDAL FUZZY CRITICAL PATH METHOD CALCULATOR

* *

ES=Earliest Start; EF=Earliest Finish; LS=Latest Start, LF=Latest Finish

* *

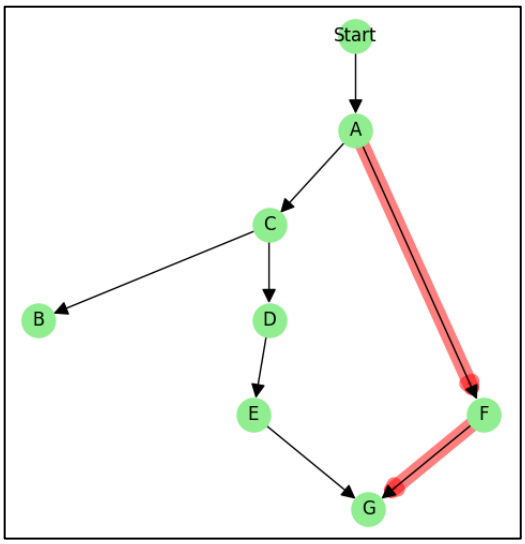
```
     DESCRIPTION ACTIVITY PREDECESSORS    F1  ... S4 DEFUZZIFI CRITICAL
       Mobilisasi        A        Start   29.0 ... 19     0.000      YES
  Pekerjaan Tanah        C            A   31.0 ... 56    39.906       NO
        Struktur        F            A  100.0 ... 19     0.000      YES
        Drainase        B            C   50.0 ... 56    39.906       NO
Perkerasan Berbutir     D            C    4.0 ... 80    62.406       NO
  Perkerasan Aspal      E            D    4.0 ... 80    62.406       NO
Pekerjaan Lain-lain     G           EF   18.0 ... 19     0.000      YES

[7 rows x 30 columns]
```

* *

Jalur Kritisnya adalah: A - F - G

Total durasi project dalam fuzzy: [147.0, 153.0, 159.0, 166.0] unit time



Gambar 5. Jalur Kritis Proyek Jembatan

Berdasarkan gambar 5 dan hasil yang diperoleh, menunjukkan bahwa A-F-G merupakan jalur kritis proyek dengan durasi normal 153-159 hari, tercepat 147 hari, dan terlambat 166 hari.

Pembahasan

Berdasarkan hasil uji coba program diperoleh beberapa analisis hasil. Berbagai macam aktivitas proyek dan durasi yang berbeda-beda menyebabkan perlunya penjadwalan atau pengaturan hubungan antar aktivitas-aktivitas secara terstruktur sehingga dapat tercapai efisiensi waktu, biaya dan tenaga. Jalur kritis proyek yaitu jalur terpanjang dalam diagram jaringan dengan jumlah *slack* (waktu longgar) paling sedikit sehingga menentukan waktu tersingkat dalam penyelesaian proyek. Pada perencanaan dan penyusunan jadwal proyek, total *slack* menunjukkan jumlah waktu yang diperkenankan suatu aktivitas boleh ditunda, tanpa mempengaruhi jadwal penyelesaian proyek secara keseluruhan.

Hasil uji coba program menggunakan data acak dengan metode random Microsoft Excel menunjukkan bahwa program sudah berjalan sesuai logika ketergantungan setiap aktivitas dengan aktivitas pendahulunya. Uji coba data acak menggunakan 12 aktivitas proyek menunjukkan sejumlah 6 aktivitas kristis dengan A - B - D - F – G – L merupakan jalur kritis proyek. Dengan demikian dapat diketahui bahwa program yang disusun berjalan sesuai dengan algoritma FCPM dan banyaknya aktivitas tidak mempengaruhi jumlah aktivitas kritis.

Pada studi kasus proyek peningkatan jalan yang telah terlaksana dengan waktu penyelesaian 120 hari, hasil uji coba program menunjukkan bahwa terdapat 5 aktivitas kritis dengan A - C - D - F - G merupakan jalur kritis proyek. Pada rencana penjadwalan, data diolah menggunakan FCPM memperoleh durasi normal 112 sampai 119 hari, dengan durasi tercepat 105 hari, dan terlama 126 hari. Proses validasi program menggunakan metode konvensional diagram jaringan, menunjukan hasil yang hampir mendekati waktu nyata penyelesaian proyek dengan berbagai opsi kemungkinan. Berdasarkan penerapan FCPM, proyek dapat terselesaikan dalam waktu 105 hari, dengan perhatian khusus pada jalur kritis agar tidak terjadi keterlambatan. Aktivitas tersebut yaitu mobilisasi, pekerjaan tanah, pekerjaan aspal, pekerjaan trotoar, dan pekerjaan lain-lain.

Pada studi kasus proyek peningkatan jembatan yang sedang berjalan dengan rencana penjadwalan menggunakan gantt chart tanpa memperhatikan logika ketergantungan aktivitas menunjukan bahwa proyek diperkirakan selesai dalam waktu 150 hari. Sedangkan pada rencana penjadwalan menggunakan FCPM diperoleh durasi normal 153 sampai 159 hari, durasi tercepat 147 hari, dan terlama 166 hari dengan perhatian khusus pada jalur kritis agar tidak terjadi keterlambatan. Hasil uji coba program yang dilakukan menunjukkan bahwa terdapat 3 aktivitas kritis yaitu mobilisasi, struktur, dan pekerjaan lain-lain dengan A – F - G merupakan lintasan kritis proyek.

Dalam rencana pengerjaan proyek pada kasus proyek peningkatan jalan dan kasus proyek peningkatan jembatan, aktivitas dapat digolongkan menjadi aktivitas seri, paralel dan terdapat juga aktivitas yang dapat dikerjakan kapan saja tanpa terpengaruh kegiatan lain. Aktivitas yang dapat dilakukan secara bersamaan (paralel) yaitu aktivitas yang memiliki aktivitas pendahulu sama, karena tidak saling ketergantungan seperti halnya pada aktivitas B, C dan E dalam pengerjaan proyek peningkatan jalan, namun aktivitas seri dapat dikerjakan jika kegiatan lain sudah diselesaikan contohnya pada aktivitas D, F dan G dalam kasus proyek jalan, dapat berjalan ketika aktvitas pendahulunya sudah selesai dilaksanakan. Pada penjadwalan FCPM, jalur kritis hanya memperhitungkan waktu sehingga tidak harus terdiri atas aktivitas yang paling penting. Jalur kritis juga dapat berubah sejalan dengan perkembangan proyek. Jika satu atau lebih aktivitas pada jalur kritis ternyata membutuhkan waktu yang lebih lama daripada yang direncanakan, maka akan menambah durasi atau umur proyek, kecuali terdapat tindakan koreksi. Pada proyek yang lebih besar dan lebih kompleks memungkinkan bahwa jalur kritis

tidak tunggal, hal tersebut dapat terjadi ketika dua atau lebih jalur memiliki durasi waktu yang sama, namun kedua kasus proyek pada penelitian ini hanya memiliki satu jalur kritis.

Penjadwalan dan pengendalian proyek merupakan faktor kunci dalam kesuksesan manajemen proyek, sebab proyek yang melebihi batas waktu proyek akan mendapatkan konsekuensi finansial dan menurunkan manfaat proyek. Keterlambatan dalam setiap aktivitas di jalur kritis akan menunda durasi proyek secara keseluruhan dan berpengaruh terhadap biaya proyek. Biaya pada proyek peningkatan jalan dan jembatan dihitung berdasarkan satuan pekerjaan dan tidak memiliki biaya *crash cost*. *Crash cost* merupakan biaya yang dikeluarkan untuk menyelesaikan kegiatan dengan waktu yang dipercepat, sehingga jika proyek selesai lebih cepat, maka penyedia jasa tidak mendapat keuntungan. Namun, pada kedua proyek ini, menurut Syarat-Syarat Umum Kontrak (SSUK) dan Syarat-Syarat Khusus Kontrak (SSKK) jika terjadi keterlambatan maka penyedia jasa akan dikenakan denda senilai 0,1% per hari dari biaya kontrak, sehingga diharapkan proyek dapat terlaksana tepat waktu.

Pada proyek peningkatan jalan, pelaksanaan proyek berjalan sesuai dengan rencana penjadwalan selama 120 hari dan tidak terjadi keterlambatan, sehingga biaya yang dikeluarkan sesuai dengan kontrak perjanjian yaitu senilai Rp3.150.000.000. Namun apabila ternyata memiliki keterlambatan, biaya denda yang dikeluarkan yaitu sebesar Rp3.150.000 per hari, sehingga jika terjadi keterlambatan sampai 126 hari biaya proyek dapat mencapai Rp3.168.900.000. Pada proyek peningkatan jembatan yang belum selesai, terdapat kemungkinan terjadi keterlambatan yang berpengaruh terhadap biaya proyek. Apabila proyek berjalan tepat waktu selama 150 hari, maka biaya yang dikeluarkan senilai Rp5.541.587.747. Namun apabila ternyata memiliki keterlambatan, biaya denda yang dikeluarkan yaitu sebesar Rp5.541.587 per hari, sehingga jika terjadi keterlambatan sampai 153 hari maka biaya penyelesaian proyek dapat mencapai Rp5.558.212.510, jika terjadi keterlambatan sampai 159 hari maka biaya dapat mencapai Rp5.591.462.037, dan jika terjadi keterlambatan sampai 166 hari maka biaya dapat mencapai Rp5.630.253.151.

Hasil penjadwalan menggunakan FCPM dapat secara efektif membantu pengguna mewujudkan hubungan antara kegiatan dalam proyek dan memiliki gagasan tentang kemajuan sepanjang durasi proyek, namun terdapat kemungkinan penundaan waktu untuk kegiatan lain diperoleh dari hasil FCPM. Nilai *slack* untuk masing-masing kegiatan juga dapat dihitung dengan mempertimbangkan pengaruhnya terhadap waktu tunda dari rangkaian kegiatan lain. Nilai *slack* merupakan bagian dari *slack total* dan mewakili waktu di mana suatu kegiatan dapat dilakukan tanpa mempengaruhi *slack* dari aktivitas berikutnya dan dapat ditunda untuk memulai tanpa mempengaruhi *slack* dari aktivitas sebelumnya. Dalam penelitian ini dikembangkan program komputer berbasis Python untuk menentukan tugas atau pekerjaan apa yang harus ditunda atau tidak dan kapan harus dimulai, namun penyajian grafis saat ini hanya menyajikan jalur kritis tanpa tampilan perhitungan. Program ini juga dikembangkan untuk melakukan analisis jalur kritis tanpa perlu menggambar diagram jaringan sehingga pada penelitian selanjutnya dapat difokuskan dalam pembuatan diagram jaringan dengan penyajian jalur kritis dan presentasi grafis yang lengkap serta mampu menyajikan prosedur perhitungan langkah demi langkah.

SIMPULAN

Rumusan masalah penjadwalan proyek secara matematis digunakan sebagai acuan dalam membuat algoritma. Pada penyusunan program FCPM, komputasi untuk mengidentifikasi jalur kritis tidak memerlukan jaringan proyek, yang diperlukan adalah pembuatan *function* pembaca data, perhitungan maju (*forward pass*), perhitungan mundur (*backward pass*), perhitungan kelonggaran waktu (*slack time*), proses defuzzifikasi dan penentuan lintasan kritis.

Program yang telah disusun akan mempermudah perhitungan. Pada program, nilai (ES1,ES2,ES3,ES4) setiap aktivitas menunjukkan estimasi waktu tercepat dimulainya *event*. Nilai (EF1,EF2,EF3,EF4) setiap aktivitas menunjukkan estimasi waktu tercepat selesainya *event*. Nilai (LS1,LS2,LS3,LS4) setiap aktivitas menunjukkan estimasi waktu paling lambat dimulainya *event*. Nilai (LF1,LF2,LF3,LF4) setiap aktivitas menunjukkan estimasi waktu paling lambat selesainya *event*. Nilai (S1,S2,S3,S4) menunjukkan estimasi kelonggaran waktu setiap aktivitas. Pada program, defuzzifikasi menunjukkan nilai *crisp* dari nilai (S1,S2,S3,S4) setiap aktivitas. Aktivitas yang memiliki nilai defuzzifikasi nol disebut aktivitas kritis.

Hasil uji coba penerapan FCPM menggunakan program Python menghasilkan *output* estimasi total durasi proyek beserta jalur kritisnya. Durasi proyek dalam bentuk (a, b, c, d) hari berarti bahwa proyek secara normal akan selesai antara b hingga c hari dengan perkiraan tercepat a hari dan toleransi keterlambatan hingga d hari. Hasil penelitian menunjukkan bahwa aktivitas mobilisasi, pekerjaan tanah, pekerjaan aspal, pekerjaan trotoar, dan pekerjaan lain-lain pada proyek peningkatan jalan merupakan aktivitas kritis. Durasi normal penyelesaian proyek adalah 112 sampai 119 hari, dengan durasi tercepat 105 hari, dan terlama 126 hari. Hasil tersebut mendekati penyelesaian secara nyata di lapangan yang berlangsung selama 120 hari. Pada proyek peningkatan jembatan, aktivitas mobilisasi, struktur, dan pekerjaan lain-lain merupakan aktivitas kritis. Durasi normal penyelesaian proyek adalah 153 sampai 159 hari, dengan durasi tercepat 147 hari, dan terlama 166 hari. Hasil tersebut mendekati rencana penjadwalan yang akan dilaksanakan selama 150 hari.

UCAPAN TERIMAKASIH

Terimakasih kepada koordinator dan seluruh Dosen Prodi Matematika yang telah memberikan ilmu dan bimbingan hingga terselesaikannya artikel ini.

DAFTAR PUSTAKA

Atin, S., & Lubis, R. (2019). Implementation of Critical Path Method in Project Planning and Scheduling. *Materials Science and Engineering*.

Bhasin, H. (2018). *Python Basics: A Self-Teaching Introduction.* United States: David Pallai.

Chan, J. (2014). *Learn Python in One Day and Learn It Well Python for Beginners with Hands-on Project The only book you need to start coding in Python immediately.* CreateSpace Independent Publishing.

Johnson, B. (2019). *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers.* Indianapolis: John Wiley & Sons.

Nasution, S. H. (1996). Metode Lintasan Kritis Kabur: Hasil yang telah dicapai. *Majalah BPPT*.

Nový, M., Nováková, J., & Waldhans, M. (2012). Project Management in Building Industry Management. *Acta Universitatis Agriculturae Et Silviculturae Mendelianae Brunensis*, 189-198.

Nowpada, R. S., Rao, P. P., & Veeramachaneni, V. S. (2010). An Analytical Method for Finding Critical Path in a Fuzzy Project Network. *Jurnal Internasional*, 5(20), 953–962.

Nowpada, R. S., Veeramachaneni, V. S., & Rao, P. P. (2010). Critical path analysis in the fuzzy project network. *Advances in fuzzy mathematics*, 5(3), 285-294.

Rani, H. A. (2016). *Manajemen Proyek Konstruksi*. Yogyakarta: Deepublish.

Taha, H. A. (2017). *Operations Research: An Introduction Tenth Edition.* United States: Pearson.

Vanhoucke, M. (2012). *Project Management with Dynamic Scheduling, 2nd Edition.* Berlin Heidelberg: Springer.