

**PENYELESAIAN *FUZZY RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM* SEBAGAI PERMASALAHAN *MULTI-KNAPSACK* (SOLVING *FUZZY RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM AS A MULTI-KNAPSACK PROBLEM* )**

SHIDKA HILDA MAGHFIROH, IRWAN ENDRAYANTO\*

**Abstract.** Scheduling is one of the crucial aspects of project management because it will determine the effectiveness of the project. Scheduling with limited resource constraints is known as the Resources Constrained Project Scheduling Problem (RCPSP). The main problem with RCPSP is constructing a schedule that produces a minimum total duration. This minimum project duration is obtained by solving the RCPSP, transformed into a Knapsack Problem sequence. In real life, each project is unique, so the duration of activity can often not be predicted precisely. The duration of activity with uncertainty aspect is known as a fuzzy number. The project schedule must consider the uncertainty aspect so that this RCPSP is expanded to become a fuzzy RCPSP. The schedule risk is used to evaluate the schedule that has been formed.

*Keywords:* Fuzzy, RCPSP, Knapsack Problem, Schedule Risk

**Abstrak.** Salah satu aspek penting dalam suatu manajemen proyek adalah penjadwalan. Efektivitas pengerjaan suatu proyek ditentukan oleh penjadwalan yang digunakan. Penjadwalan yang memiliki keterbatasan sumber daya dikenal sebagai *Resources Constrained Project Scheduling Problem* (RCPSP). Permasalahan utama pada RCPSP adalah konstruksi jadwal yang akan menghasilkan minimum total durasi dengan mentransformasikan ke dalam barisan *Knapsack Problem*. Pada kehidupan nyata, setiap proyek bersifat unik sehingga durasi aktivitasnya seringkali tidak bisa diprediksi secara pasti. Durasi untuk aktivitas yang memiliki aspek ketidakpastian dapat dituliskan sebagai bilangan *fuzzy*, sehingga RCPSP ini diperluas menjadi *fuzzy RCPSP*. Selanjutnya untuk menghitung ukuran performansi suatu penjadwalan, maka digunakan *schedule risk* yang dibangun untuk mengevaluasi jadwal yang telah terbentuk.

*Kata-kata kunci:* Fuzzy, RCPSP, Knapsack Problem, Schedule Risk

## 1. PENDAHULUAN

Masalah penjadwalan adalah salah satu permasalahan yang sering dihadapi dalam manajemen proyek. Penjadwalan proyek berperan aktif dalam menentukan waktu mulai serta mengalokasikan sumber daya untuk setiap aktivitas. Semakin banyak aktivitas pada suatu proyek, maka tingkat kompleksitas penjadwalannya akan semakin tinggi. Penjadwalan proyek yang memperhatikan kendala keterbatasan sumber daya dikenal sebagai *Resources-Constrained Project Scheduling Problem* (RCPSP). RCPSP memiliki dua kendala utama, yaitu kendala hubungan antar aktivitas dan keterbatasan sumber daya.

RCPSP juga dapat dipandang sebagai masalah optimisasi, yaitu pencarian jadwal yang menghasilkan total durasi proyek minimum. Meminimumkan total durasi proyek secara keseluruhan dapat diperoleh dengan cara meminimumkan total durasi beberapa aktivitas secara bertahap. Sehingga perlu adanya suatu mekanisme agar pilihan aktivitas yang dikerjakan masing-masing tahap merupakan kombinasi yang tepat. Mekanisme ini dapat dipandang sebagai *Knapsack Problem* oleh [1], yaitu pemilihan sejumlah aktivitas untuk dikerjakan agar nilai fungsi objektifnya maksimum dengan kendala sumber daya yang terbatas. Mekanisme ini terus diulang hingga semua aktivitas selesai dikerjakan. Masalah penjadwalan ini dapat ditransformasi menjadi barisan *Knapsack Problem* oleh [6].

Pada kehidupan nyata, setiap proyek bersifat unik. Durasi aktivitas yang diestimasi oleh manager proyek seringkali tidak dapat dilakukan secara pasti sehingga memiliki aspek ketidakpastian. Estimasi durasi aktivitas, waktu mulai, dan waktu selesai proyek yang memiliki aspek ketidakpastian ini akan dinyatakan dengan bilangan *fuzzy*. Masalah penjadwalan ini kemudian diperluas menjadi *Fuzzy RCPSP*. Biasanya permasalahan *fuzzy RCPSP* diselesaikan dengan metode pencarian lain seperti *Fuzzy Beam Search Algorithm*, lihat [10]. Namun pada permasalahan kali ini algoritma baru akan dikembangkan untuk menyelesaikan *fuzzy RCPSP* dengan pengembangan Algoritma *Knapsack* pada [6] dan memperhatikan durasi dalam bilangan *fuzzy*.

Pada penjadwalan proyek dengan durasi *fuzzy*, *schedule risk* digunakan sebagai ukuran perfomansi dari jadwal yang telah terbentuk oleh [10]. *Schedule risk* adalah suatu ukuran untuk mengetahui kemungkinan proyek yang dikerjakan berdasarkan jadwal tersebut akan selesai lebih lama dari batas waktu yang ditetapkan di awal. Jadwal yang baik tentu akan menghasilkan nilai *schedule risk* yang kecil.

## 2. FORMULASI *RESOURCES-CONSTRAINED PROJECT SCHEDULING PROBLEM*

Pada bagian ini dibahas tentang formulasi *Resources-Constrained Project Scheduling Problem* (RCPSP) serta asumsi dan parameternya.

**2.1. Asumsi dan Parameter.** Asumsi yang digunakan pada masalah RCPSP adalah:

- (1) Sumber daya yang digunakan jumlahnya tetap untuk setiap waktu, misalnya mesin, pekerja dan sebagainya.

- (2) Masing-masing aktivitas membutuhkan sumber daya dalam jumlah konstan selama durasi aktivitas tersebut.

Sedangkan parameter yang digunakan adalah:

$\mathcal{A}$	Himpunan aktivitas, $\mathcal{A} = \{0, 1, 2, \dots, n + 1\}$
$M$	Matriks <i>precedence relations</i> , dengan $i, j \in \mathbb{A}$ berlaku
	$M_{ij} = \begin{cases} 1 & \text{jika aktivitas } i \text{ adalah pendahulu bagi aktivitas } j \\ 0 & \text{lainnya.} \end{cases}$
$Pred_i$	Semua pendahulu untuk aktivitas $i$
$Suc_i$	Semua suksesor untuk aktivitas $j$
$d_i$	Durasi aktivitas $i$
$u_i$	Bobot aktivitas $i$
$\mathcal{K}$	Himpunan semua tipe sumber daya
$r_{i,k}$	Sumber daya tipe $k$ yang dibutuhkan oleh aktivitas $i$
$R_k$	Sumber daya tipe $k$ yang tersedia
$b$	Waktu yang diperbolehkan untuk memulai proyek ( <i>ready-time</i> )
$e$	Batas waktu yang direncanakan untuk menyelesaikan proyek ( <i>deadline</i> )

**2.2. Formulasi Matematika RCPSP.** Misalkan  $f$  merupakan himpunan waktu selesai setiap aktivitas,  $f = \{f_i | i \in \mathcal{A}\}$  dengan  $f_i$  adalah waktu selesai aktivitas  $i$ .  $\mathcal{A}(t) = \{i \in \mathcal{A} | f_i - d_i \leq t < f_i\}$  merupakan himpunan aktivitas yang aktif pada saat  $t$ . Formulasi RCPSP untuk meminimumkan total durasi proyek adalah sebagai berikut [5]

Meminimumkan

$$f_{n+1} \quad (2.1)$$

terhadap kendala

$$f_h \leq f_i - d_i \quad i = 1, 2, \dots, n + 1; h \in Pred_i \quad (2.2)$$

$$\sum_{i \in \mathcal{A}(t)} r_{i,k} \leq R_k \quad \forall k \in \mathcal{K}; t \geq 0 \quad (2.3)$$

$$f_0 \geq b. \quad (2.4)$$

Persamaan (2.1) merupakan fungsi objektif pada permasalahan ini, yaitu waktu selesai aktivitas *dummy* terakhir yang menyatakan waktu selesai proyek. Persamaan (2.2) merupakan kendala relasi antar aktivitas. Kendala ini didasarkan pada aturan bahwa setiap aktivitas baru boleh dikerjakan apabila semua aktivitas pendahulunya selesai. Persamaan (2.3) merupakan kendala keterbatasan sumber daya. Kendala ini menyatakan bahwa jumlah sumber daya yang digunakan aktivitas aktif tidak boleh melebihi dari jumlah sumber daya yang tersedia. Hal tersebut juga berlaku untuk seluruh tipe sumber daya. Kendala pada Persamaan (2.4) menunjukkan bahwa proyek tersebut harus dilakukan pada waktu yang diperbolehkan untuk memulai proyek.

### 3. FORMULASI FUZZY RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM

*Fuzzy Resource-Constrained Project Scheduling Problem* atau *fuzzy RCPSP* merupakan perluasan dari *Resource-Constrained Project Scheduling Problem*. *Fuzzy RCPSP* memperhatikan aspek ketidakpastian pada durasi aktivitas yang dianggap sebagai bilangan *fuzzy*. Ketidakpastian durasi aktivitas dinotasikan sebagai  $\tilde{d}$  dengan  $\tilde{d} = (t_1, t_2, t_3, t_4)$ .

$$\mu_{\tilde{d}}(x) = \begin{cases} 0 & , x < t_1, x > t_4 \\ \frac{x-t_1}{t_2-t_1} & , t_1 \leq x < t_2 \\ 1 & , t_2 \leq x \leq t_3 \\ \frac{t_4-x}{t_4-t_3} & , t_3 < x \leq t_4, \end{cases}$$

dengan  $t_1 < t_2 < t_3 < t_4$ .

*Fuzzy ready time* yang dinotasikan sebagai  $\tilde{b}$  merupakan kemungkinan waktu mulai yang memungkinkan untuk suatu proyek, sedangkan *fuzzy deadline* yang dinotasikan sebagai  $\tilde{e}$  merupakan kemungkinan waktu selesai yang diharapkan untuk suatu proyek. Berbeda dengan durasi aktivitas, *fuzzy ready time* dan *fuzzy deadline* merupakan preferensi dari manager proyek. Sebagai contoh, misalkan seorang manager proyek ingin menyelesaikan proyek sebelum waktu  $e_1$  namun tidak lebih dari waktu  $e_2$  agar tidak menunda masuknya produk ke pasar. Sehingga *deadline* proyek yang dipilih dapat direpresentasikan sebagai bilangan *fuzzy* trapesium  $\tilde{e} = (e_1, e_2, e_3, e_4)$ , untuk  $e_1 < e_2 < e_3 < e_4$ . Dengan cara yang sama, *ready-time* proyek yang dipilih dapat direpresentasikan sebagai bilangan *fuzzy* trapesium  $\tilde{b} = (b_1, b_2, b_3, b_4)$ , dengan  $b_1 < b_2 < b_3 < b_4$ .

Formulasi matematika *fuzzy RCPSP* yang terdiri dari  $n$  aktivitas dituliskan dalam bentuk berikut

Meminimumkan

$$\widetilde{f_{n+1}} \quad (3.1)$$

dengan kendala

$$\widetilde{f}_h \leq \widetilde{f}_i - \widetilde{d}_i \quad i = 1, 2, \dots, n+1; h \in \text{Pred}_i \quad (3.2)$$

$$\sum_{i \in A(t)} r_{i,k} \leq R_k \quad \forall k \in \mathcal{K}; t \geq 0 \quad (3.3)$$

$$\widetilde{f}_0 \geq \tilde{b} \quad (3.4)$$

dengan

- $\widetilde{f}_i$  adalah waktu selesai dalam *fuzzy* untuk aktivitas  $i$
- $\widetilde{d}_i$  adalah durasi dalam *fuzzy* untuk aktivitas  $i$
- $\tilde{b}$  adalah *fuzzy ready-time*

Formulasi masalah *fuzzy RCPSP* tersebut tidak bisa diselesaikan secara langsung menggunakan metode pemrograman linear pada umumnya, karena kendala pada persamaan (3.2) memiliki  $A(t)$  yang merupakan variabel dan nilainya akan berubah seiring dengan berjalannya  $t$ . Sehingga diperlukan adanya mekanisme atau cara lainnya yang dapat menyelesaikan permasalahan penjadwalan tersebut.

#### 4. PENYELESAIAN *FUZZY RCPSP* SEBAGAI BARISAN *MULTI KNAPSACK PROBLEM*

Penjadwalan yang baik akan menghasilkan total durasi proyek yang efektif. Total durasi proyek minimum akan tercapai apabila dapat memilih aktivitas yang dikerjakan setiap waktu secara tepat. Pilihan yang tepat dalam hal ini dibagi menjadi dua kasus, yaitu

- Apabila pada waktu tertentu terdapat beberapa aktivitas yang diperbolehkan untuk dikerjakan secara bersamaan dan jumlah sumber daya yang tersedia masih mencukupi, maka tidak boleh ada aktivitas yang mengalami penundaan.
- Apabila pada waktu tertentu terdapat sejumlah aktivitas yang diperbolehkan untuk dikerjakan secara bersamaan namun total sumber daya yang tersedia tidak mencukupi, maka harus ada aktivitas yang ditunda. Penentuan aktivitas mana yang harus dikerjakan terlebih dahulu atau ditunda ini akan dilihat dari bobot aktivitas tersebut.

Berdasarkan *Ranked Position Weight Method* [9], bobot suatu aktivitas didefinisikan sebagai durasi aktivitas ditambah dengan durasi seluruh aktivitas yang mengikutinya. Misal  $\tilde{w}_i$  adalah bobot aktivitas  $i$ , maka

$$\tilde{w}_i = \tilde{d}_i + \sum_{j \in S_{uc_i}} \tilde{d}_j \quad (4.1)$$

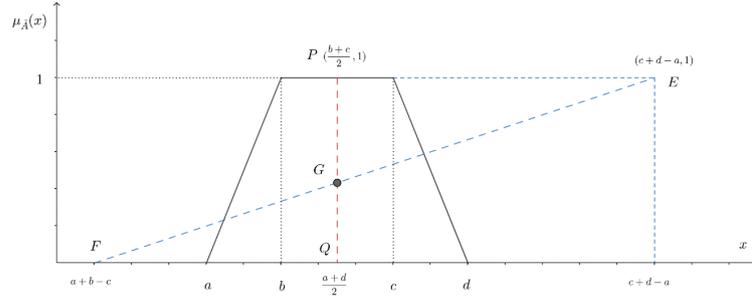
Bobot dari aktivitas menunjukkan tingkat kepentingan aktivitas relatif terhadap aktivitas lainnya. Aktivitas dengan bobot yang lebih besar memiliki prioritas lebih tinggi untuk dikerjakan terlebih dahulu daripada aktivitas lain yang memiliki bobot lebih kecil. Pemilihan aktivitas mana yang harus dikerjakan atau ditunda pada setiap waktu dapat dipandang sebagai permasalahan *Knapsack*. Tujuan permasalahan *Knapsack* tersebut adalah untuk memaksimalkan jumlah bobot aktivitas yang dikerjakan dengan kendala keterbatasan sumber daya.

**4.1. Formulasi Multi Knapsack Problem dengan Ranking *Fuzzy*.** Pemilihan aktivitas yang harus dikerjakan atau ditunda pada setiap waktunya dapat dipandang sebagai permasalahan *Multi Knapsack*. Karena bobot masing-masing aktivitas berupa bilangan *fuzzy*, maka perlu adanya ranking *fuzzy* untuk merubah menjadi bobot *non-fuzzy*.

**Ranking Bilangan Fuzzy.** Diambil  $N(\mathbb{R})$  sebagai himpunan dari semua bilangan *fuzzy* pada  $\mathbb{R}$  dan  $A, B \in N(\mathbb{R})$ . Fungsi  $F : N(\mathbb{R}) \Rightarrow \mathbb{R}$  disebut fungsi ranking atau index ranking dengan  $F(A) \leq F(B)$  ekuivalen  $A(\leq)B$ .

**Definisi 4.1.** [11] Diberikan bilangan *fuzzy* trapesium  $\tilde{A} = (a, b, c, d)$ , dan fungsi keanggotaan sebagai berikut

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & , x < a, x > d \\ \frac{x-a}{b-a} = f_{\tilde{A}}^L(x) & , a \leq x < b \\ 1 & , b \leq x \leq c \\ \frac{d-x}{d-c} = f_{\tilde{A}}^R(x) & , c < x \leq d. \end{cases}$$



GAMBAR 1. Centroid bilangan fuzzy trapesium  $\tilde{A}$

Nilai centroid untuk setiap  $\tilde{A}$  dinyatakan sebagai berikut

$$\begin{aligned} \text{Centroid}(\tilde{A}) &= \frac{\int_a^b x f_{\tilde{A}}^L(x) dx + \int_b^c x dx + \int_c^d x f_{\tilde{A}}^R(x) dx}{\int_a^b f_{\tilde{A}}^L(x) dx + \int_b^c 1 dx + \int_c^d f_{\tilde{A}}^R(x) dx} \\ &= \frac{c^2 + d^2 - (a^2 + b^2) + cd - ab}{3[(c+d) - (a+b)]} \end{aligned}$$

Sehingga didapatkan formulasi Multi *Knapsack Problem* untuk penjadwalan proyek tersebut sebagai berikut

Memaksimumkan

$$Z = \sum_{i \in E} u_i x_i \quad (4.2)$$

dengan kendala

$$\sum_{i \in E} r_{i,k} x_i \leq RS_k \quad \forall k \in \mathcal{K} \quad (4.3)$$

$$x_i \in \{0, 1\} \quad \forall i \in E. \quad (4.4)$$

Dengan

- $u_i$  merupakan bobot *non-fuzzy* untuk aktivitas  $i$  sebagai centroid dari  $\tilde{w}_i$ . Misal  $\tilde{w}_i = (a_i, b_i, c_i, d_i)$ , maka

$$u_i = \frac{c_i^2 + d_i^2 - (a_i^2 + b_i^2) + c_i d_i - a_i b_i}{3[(c_i + d_i) - (a_i + b_i)]},$$

- $E$  adalah *eligible activity*, yaitu aktivitas yang mungkin untuk dikerjakan,
- $r_{i,k}$  adalah sumber daya tipe  $k$  yang dibutuhkan oleh aktivitas  $i$
- $RS_k$  adalah sumber daya tipe  $k$  yang sedang tidak digunakan pada saat itu.

Variabel keputusan permasalahan di atas adalah aktivitas  $i$  dipilih untuk dikerjakan jika  $x_i = 1$  atau aktivitas  $i$  ditunda untuk dikerjakan apabila  $x_i = 0$ . *Eligible activity* akan diupdate pada setiap iterasi Proses ini akan diulang sehingga semua aktivitas selesai dikerjakan. Penyelesaian masalah penjadwalan tersebut dapat dilakukan dengan melakukan transformasi *fuzzy* RCPSP menjadi barisan *Knapsack Problem*. Selanjutnya akan diberikan Algoritma Penyelesaian untuk *Fuzzy* RCPSP.

**4.2. Algoritma Penyelesaian *Fuzzy* RCPSP.** *Fuzzy* RCPSP memperhatikan aspek ketidakpastian pada durasi aktivitas yang dianggap sebagai bilangan *fuzzy*. Selain durasi aktivitas, manajer proyek juga untuk menentukan waktu mulai dan waktu yang diharapkan untuk menyelesaikan proyek dalam bilangan *fuzzy* yang disebut sebagai *fuzzy ready-time* dan *fuzzy deadline*. Sehingga untuk durasi, *ready time* dan *deadline* dalam bilangan *fuzzy*, dapat dituliskan pada parameter dan variabel sebagai berikut

**Parameter**

$\mathcal{A}$	Himpunan aktivitas, $\mathcal{A} = \{0, 1, 2, \dots, n + 1\}$
$M$	Matriks <i>precedence relations</i>
$Suc_i$	Semua suksesor untuk aktivitas $j$
$\tilde{d}_i$	Durasi <i>fuzzy</i> aktivitas $i$
$u_i$	Bobot aktivitas $i$
$\mathcal{K}$	Himpunan semua tipe sumber daya
$r_{i,k}$	Sumber daya tipe $k$ yang dibutuhkan oleh aktivitas $i$
$R_k$	Sumber daya tipe $k$ yang tersedia
$\tilde{b}$	<i>Fuzzy ready-time</i> .

**Variabel**

$\mathcal{F}$	Himpunan aktivitas yang sudah selesai dikerjakan
$F_T$	Aktivitas terakhir yang selesai pada setiap iterasi
$\mathcal{P}$	Himpunan aktivitas yang sedang dalam proses
$\mathcal{U}$	Himpunan aktivitas yang belum selesai ( <i>unfinished</i> )
$\mathcal{N}$	Himpunan aktivitas yang belum selesai namun belum diperbolehkan untuk dikerjakan ( <i>not eligible</i> )
$E$	Himpunan aktivitas yang mungkin untuk dikerjakan ( <i>eligible</i> )
$RS_k$	Jumlah sumber daya tipe $k$ yang tidak sedang digunakan
$\tilde{s}_i$	Waktu mulai aktivitas $i$
$\tilde{f}_i$	Waktu selesai aktivitas $i$ .

Pada *fuzzy* RCPSP, pemilihan aktivitas yang harus dikerjakan atau ditunda pada setiap waktunya dapat dipandang sebagai *Multi Knapsack Problem* dengan tujuan memaksimalkan jumlah bobot aktivitas yang terpilih untuk dikerjakan dan dengan kendala keterbatasan sumber daya. Pemilihan menggunakan *Multi Knapsack Problem* tersebut diteruskan hingga semua aktivitas selesai dikerjakan. Sehingga metode penyelesaian *fuzzy* RCPSP tersebut dapat diselesaikan dengan barisan *Multi Knapsack Problem* sebagai berikut

**Algoritma**

(1) **Inisialisasi**

$$\mathcal{F} = F_T = \{0\}$$

$$\mathcal{P} = \mathcal{U} = \mathcal{N} = E = \emptyset$$

$$RS_k = R_k \quad \forall k \in \mathcal{K}$$

$$\tilde{s}_i = \tilde{f}_i = (0, 0, 0, 0) \quad \forall i \in \mathcal{A} \setminus \{0\}$$

$$\tilde{s}_0 = \tilde{f}_0 = \tilde{b}$$

$u_{n+1} \neq 0$  (agar *Knapsack Problem* dapat diselesaikan, bobot tidak boleh nol).

(2) **Mencari aktivitas yang mungkin untuk dikerjakan**

Aktivitas yang belum selesai dikerjakan adalah himpunan semua aktivitas dikurangi aktivitas yang sudah selesai.

$$\mathcal{U} = \mathcal{A} \setminus \mathcal{F}.$$

Aktivitas yang belum selesai namun belum diperbolehkan untuk dikerjakan adalah aktivitas yang merupakan suksesor dari aktivitas lain yang juga belum selesai.

$$\mathcal{N} = \bigcup_{j \in \mathcal{U}} \text{Suc}_j.$$

Dengan demikian, aktivitas yang mungkin atau diperbolehkan untuk dikerjakan adalah

$$E = \mathcal{U} \setminus \{\mathcal{N} \cup \mathcal{P}\}.$$

(3) **Pengalokasian sumber daya sebagai *Knapsack Problem***

Setiap aktivitas yang mungkin atau diperbolehkan untuk dikerjakan belum tentu dapat dilaksanakan karena adanya keterbatasan sumber daya. Pemilihan aktivitas mana saja yang akan dikerjakan atau ditunda dapat dianggap sebagai *Knapsack Problem*, yaitu

Memaksimalkan

$$Z = \sum_{i \in E} u_i x_i \quad (4.5)$$

dengan kendala

$$\sum_{i \in E} r_{i,k} x_i \leq RS_k \quad \forall k \in \mathcal{K} \quad (4.6)$$

$$x_i \in \{0, 1\} \quad \forall i \in E. \quad (4.7)$$

Aktivitas  $i$  dikerjakan apabila  $x_i = 1$ , sebaliknya aktivitas  $i$  ditunda apabila  $x_i = 0$ .

(4) **Menentukan aktivitas yang sedang dalam proses**

$$\mathcal{P} \leftarrow \mathcal{P} \cup \{i | x_i = 1\}.$$

- (5) **Menentukan waktu mulai dan waktu selesai aktivitas yang sedang dalam proses**

Untuk  $\forall i \in \mathcal{P}$ , jika  $\tilde{s}_i = \tilde{f}_i = (0, 0, 0, 0)$ , maka

$$\tilde{s}_i = \widetilde{f_{F_T}}$$

$$\tilde{f}_i = \tilde{s}_i \oplus \tilde{d}_i.$$

- (6) **Menentukan aktivitas yang sudah selesai**

Misal  $\tilde{f}_i = (f_{1i}, f_{2i}, f_{3i}, f_{4i})$ , maka

$$\mathcal{F} \leftarrow \mathcal{F} \cup \{j | f_{4j} = \min\{f_{4i}\}; i \in \mathcal{P}\}$$

dan aktivitas terakhir yang sudah selesai adalah

$$F_T = j.$$

- (7) **Kriteria penghentian iterasi**

Jika  $\mathcal{F} = \mathcal{A}$ , maka iterasi selesai. Jika tidak, lanjutkan ke langkah 8.

- (8) **Menentukan aktivitas yang sedang dalam proses**

Perbarui aktivitas yang sedang dalam proses dengan menghapus aktivitas yang sebelumnya dalam proses namun sekarang sudah selesai.

$$\mathcal{P} \leftarrow \mathcal{P} \setminus \{\mathcal{P} \cap \mathcal{F}\}.$$

- (9) **Menentukan jumlah sumber daya yang tidak sedang digunakan**

$$RS_k = R_k - \sum_{i \in \mathcal{P}} r_{i,k}.$$

- (10) **Ulangi dari langkah 2**

Setelah mendapatkan algoritma untuk penyelesaian *fuzzy* RCPSP, perlu adanya perhitungan untuk resiko dari jadwal yang terbentuk. Analisa resiko jadwal tersebut dilakukan dengan menghitung *schedule risk*.

## 5. SCHEDULE RISK

Sebelum dijelaskan mengenai *schedule risk* sebagai ukuran performansi atas jadwal yang telah terbentuk, akan dijelaskan terlebih dahulu mengenai teori *Fuzzy* untuk menghitung *schedule risk*.

**5.1. Teori *Fuzzy* untuk Menghitung *Schedule Risk*.** Proyek dengan durasi aktivitas berupa bilangan *fuzzy* akan menghasilkan jadwal dengan total durasi proyek yang juga mengandung aspek ketidakpastian sehingga memiliki risiko yang disebut *schedule risk*. Selanjutnya akan diberikan beberapa konsep mengenai himpunan dan bilangan *fuzzy* digunakan untuk menghitung besar *schedule risk* tersebut.

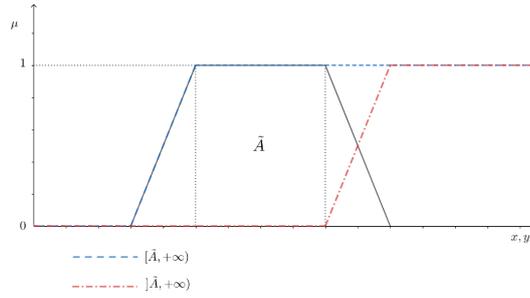
5.1.1. **Himpunan yang dibatasi oleh himpunan *fuzzy*.** Misalkan  $\tilde{A}$  adalah bilangan *fuzzy*. Empat jenis himpunan yang dibatasi oleh bilangan *fuzzy*  $\tilde{A}$  didefinisikan sebagai berikut.

- Himpunan bilangan yang mungkin lebih besar atau sama dengan (*possibly greater than or equal to*)  $\tilde{A}$  dinotasikan dengan  $[\tilde{A}, +\infty)$ .

$$\mu_{[\tilde{A}, +\infty)}(y) = \sup_{x \leq y} \mu_{\tilde{A}}(x). \quad (5.1)$$

- Himpunan bilangan yang perlu lebih besar dari (*necessarily greater than*)  $\tilde{A}$  dinotasikan dengan  $] \tilde{A}, +\infty)$ .

$$\mu_{] \tilde{A}, +\infty)}(y) = \inf_{x \geq y} (1 - \mu_{\tilde{A}}(x)) = \inf_{x \geq y} (\mu_{\tilde{A}'}(x)). \quad (5.2)$$



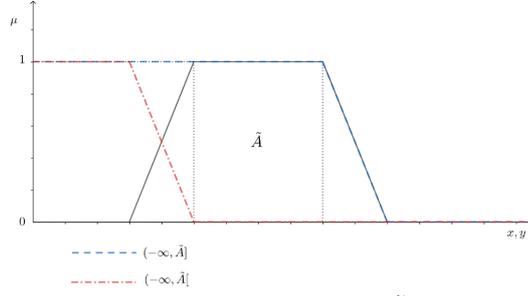
GAMBAR 2. Himpunan *fuzzy*  $[\tilde{A}, +\infty)$  dan  $] \tilde{A}, +\infty)$

- Himpunan bilangan yang mungkin lebih kecil atau sama dengan (*possibly smaller than or equal to*)  $\tilde{A}$  dinotasikan dengan  $(-\infty, \tilde{A}]$ .

$$\mu_{(-\infty, \tilde{A}]}(y) = \sup_{x \geq y} \mu_{\tilde{A}}(x). \quad (5.3)$$

- Himpunan bilangan yang perlu lebih kecil dari (*necessarily smaller than*)  $\tilde{A}$  dinotasikan dengan  $(-\infty, \tilde{A}[$ .

$$\mu_{(-\infty, \tilde{A}[}(y) = \inf_{x \leq y} (1 - \mu_{\tilde{A}}(x)) = \inf_{x \leq y} (\mu_{\tilde{A}'}(x)). \quad (5.4)$$

GAMBAR 3. Himpunan *fuzzy*  $(-\infty, \tilde{A}]$  dan  $(-\infty, \tilde{A}[$ 

**5.2. Ukuran Possibility dan Necessity.** Ukuran *possibility* dan ukuran *necessity* digunakan untuk membandingkan dua bilangan *fuzzy*. Misal diberikan dua bilangan *fuzzy*  $\tilde{M}$  dan  $\tilde{N}$  serta suatu parameter  $x$  yang dibatasi oleh  $\tilde{M}$ . Ukuran *possibility* dan ukuran *necessity* yang menyatakan bahwa " $\tilde{M}$  lebih besar atau sama dengan  $\tilde{N}$ " didefinisikan sebagai berikut.

$$PG(\tilde{M}, \tilde{N}) = \sup_x \min(\mu_{\tilde{M}}(x), \mu_{[\tilde{N}, +\infty)}(x)), \quad (5.5)$$

$$NG(\tilde{M}, \tilde{N}) = \inf_x \max(1 - \mu_{\tilde{M}}(x), \mu_{[\tilde{N}, +\infty)}(x)). \quad (5.6)$$

Persamaan (5.5) menyatakan *grade of possibility* dari pernyataan " $\tilde{M}$  lebih besar atau sama dengan  $\tilde{N}$ ". Nilai tersebut merupakan indeks yang mengestimasi kemungkinan terbesar kejadian " $\tilde{M} \geq \tilde{N}$ " akan terjadi. Sedangkan persamaan (5.6) menyatakan *grade of necessity* dari pernyataan " $\tilde{M}$  lebih besar atau sama dengan  $\tilde{N}$ ". Nilai tersebut merupakan indeks yang mengestimasi kemungkinan terkecil kejadian " $\tilde{M} \geq \tilde{N}$ " akan terjadi.

Hal yang serupa untuk ukuran *possibility* dan ukuran *necessity* yang menyatakan bahwa " $\tilde{M}$  lebih besar  $\tilde{N}$ " didefinisikan sebagai berikut.

$$PSG(\tilde{M}, \tilde{N}) = \sup_x \min(\mu_{\tilde{M}}(x), \mu_{] \tilde{N}, +\infty)}(x)) \quad (5.7)$$

$$NSG(\tilde{M}, \tilde{N}) = \inf_x \max(1 - \mu_{\tilde{M}}(x), \mu_{] \tilde{N}, +\infty)}(x)) \quad (5.8)$$

Persamaan (5.7) menyatakan *grade of possibility* dari pernyataan " $\tilde{M}$  lebih besar dari  $\tilde{N}$ ". Nilai tersebut merupakan indeks yang mengestimasi kemungkinan terbesar kejadian " $\tilde{M} > \tilde{N}$ " akan terjadi. Sedangkan persamaan (5.8) menyatakan *grade of necessity* dari pernyataan " $\tilde{M}$  lebih besar dari  $\tilde{N}$ ". Nilai tersebut merupakan indeks yang mengestimasi kemungkinan terkecil kejadian " $\tilde{M} > \tilde{N}$ " akan terjadi.

5.3. **Schedule Risk.** Suatu ukuran performansi digunakan untuk mengukur tingkat efektivitas dari jadwal proyek dengan durasi *fuzzy*. Ukuran performansi ini disebut *schedule risk*. *Schedule risk* mengukur besarkemungkinan pengerjaan proyek akan selesai lebih lama dari batas waktu yang telah ditentukan di awal.

**Definisi 5.1.** [10] Misalkan  $\tilde{b}$  dan  $\tilde{e}$  berturut-turut adalah *fuzzy ready-time* dan *fuzzy deadline* dari suatu proyek. Suatu jadwal  $s$  memiliki *schedule risk*  $SR(s)$  yang didefinisikan sebagai berikut

$$SR(s) = \beta \times PSG(\tilde{D}(s), \tilde{e} - \tilde{b}) + (1 - \beta) \times NSG(\tilde{D}(s), \tilde{e} - \tilde{b}) \quad (5.9)$$

dengan  $\tilde{D}(s)$  adalah total durasi pengerjaan proyek berdasarkan jadwal  $s$ .

Perhatikan bahwa formula untuk menghitung *schedule risk* pada Persamaan (3.24) identik dengan Persamaan (3.25).  $SR(s)$  mengukur kemungkinan durasi proyek yang dikerjakan berdasarkan jadwal  $s$  lebih besar daripada selisih antara *deadline* dan *ready-time* yang sudah ditentukan. Untuk menghitung *schedule risk*, nilai indeks optimistik-pesimistik  $\beta$  yang digunakan ditentukan oleh manajer proyek. Nilai  $\beta$  yang kurang dari 0.5 menunjukkan pesimistik sedangkan nilai  $\beta$  yang lebih dari 0.5 menunjukkan optimistik.

## 6. PENYELESAIAN MASALAH PENJADWALAN PROYEK ELEKTRONIK

Proyek pengembangan produk menjadi perhatian utama untuk sebagian industri pada beberapa tahun terakhir. Proyek pengembangan produk yang digunakan untuk simulasi kali ini adalah data proyek pengembangan produk elektronik yang diambil dari paper [10] yang terdiri dari 51 aktivitas. Sumber daya atau pekerja yang digunakan dibagi menjadi empat tipe, yaitu  $r_1$  adalah teknisi sistem,  $r_2$  adalah teknisi *software*,  $r_3$  adalah teknisi *hardware*, dan  $r_4$  adalah teknisi pendukung. *Ready-time* proyek ini adalah  $(0, 0, 0, 0)$  dan *deadline* adalah  $(220, 245, 275, 300)$ . Sumber daya yang tersedia untuk setiap tipe yaitu  $(R_1, R_2, R_3, R_4) = (5, 4, 5, 4)$ .

Formulasi matematika Fuzzy RCPSP untuk proyek tersebut adalah Meminimumkan

$$\widetilde{f_{n+1}} \quad (6.1)$$

dengan kendala

$$\begin{aligned} \widetilde{f}_h &\leq \widetilde{f}_i - \widetilde{d}_i & i = 1, 2, \dots, 52; h \in Pred_i \\ \sum_{i \in A(t)} r_{i,k} &\leq R_k & \forall k \in \{1, 2, 3, 4\}; t \geq 0 \\ \widetilde{f}_0 &\geq \tilde{b} \end{aligned} \quad (6.2)$$

dengan

- $\widetilde{f}_i$  adalah waktu selesai dalam *fuzzy* untuk aktivitas  $i$
- $\widetilde{d}_i$  adalah durasi dalam *fuzzy* untuk aktivitas  $i$
- $\tilde{b}$  adalah *fuzzy ready-time*.

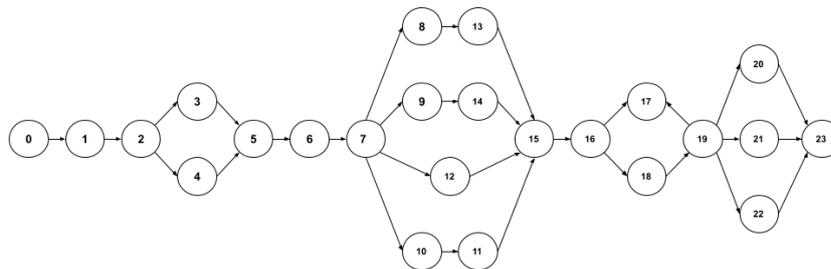
Deskripsi lengkap untuk kebutuhan pekerja untuk masing-masing tipe dan durasi proyek ini disajikan pada Tabel 1 berikut.

TABEL 1. Data Proyek Pengembangan Elektronik.

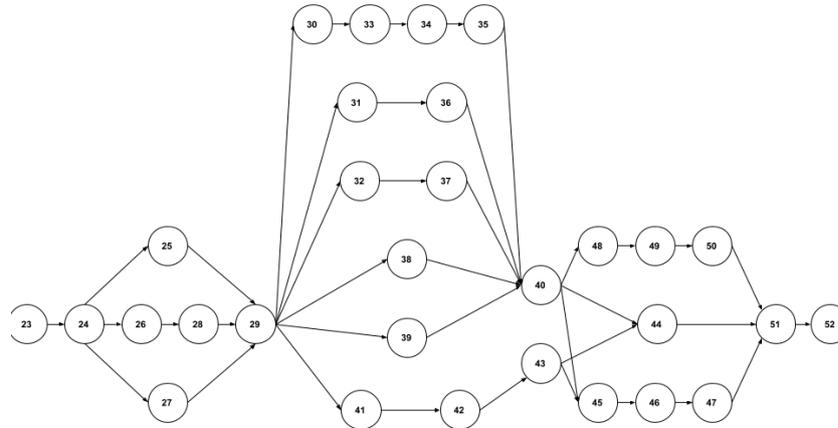
No	Deskripsi Aktivitas	$r_1$	$r_2$	$r_3$	$r_4$	Durasi	Suksesor
1	Akuisisi permintaan pelanggan	3	0	0	0	(5, 6, 7, 8)	2
2	Eksplorasi konsep	5	3	3	2	(10, 12, 13, 14)	3, 4
3	Perancangan arsitektur sistem	4	2	4	2	(4, 5, 6, 7)	5
4	Perancangan desan dan kebutuhan sistem	4	2	2	2	(7, 8, 9, 10)	5
5	Membuat rancangan spesifikasi sistem	5	0	0	0	(2, 4, 5, 6)	6
6	<i>Review</i> kebutuhan sistem	4	2	2	2	(3, 4, 5, 6)	7
7	Mengalokasikan kebutuhan sistem ke <i>hardware</i> dan <i>software</i>	4	3	3	2	(2, 3, 4, 5)	8, 9, 10, 12
8	Membuat rancangan kebutuhan <i>software</i>	0	4	2	0	(2, 3, 4, 5)	13
9	Membuat rancangan spesifikasi pengembangan <i>hardware</i>	2	1	5	3	(3, 4, 5, 6)	14
10	Finalisasi spesifikasi sistem	4	0	0	0	(4, 5, 6, 7)	11
11	Membuat dokumentasi desain sistem	5	0	0	0	(7, 9, 10, 11)	15
12	Membuat konsep pengujian sistem	3	0	0	1	(3, 4, 5, 6)	15
13	<i>Review</i> kebutuhan <i>software</i>	3	2	0	1	(1, 2, 3, 4)	15
14	<i>Review</i> spesifikasi <i>hardware</i>	3	0	3	2	(1, 2, 3, 4)	15
15	<i>Review</i> desain sistem	4	2	2	2	(1, 2, 3, 4)	16
16	Menganalisis aliran operasional dan fungsional	5	2	2	0	(3, 4, 5, 6)	17, 18
17	Mendefinisikan <i>interface</i> fungsional <i>hardware</i> dan <i>software</i>	5	2	4	2	(4, 5, 6, 7)	19
18	Partisi <i>hardware</i> dan <i>software</i>	4	2	4	3	(4, 5, 6, 7)	19
19	Menganalisis kebutuhan <i>interface</i>	5	2	3	4	(5, 6, 7, 8)	20, 21, 22
20	Mengalokasikan desain <i>software</i> dan <i>hardware</i> tingkatan yang lebih rendah	4	0	0	0	(5, 7, 8, 9)	23
21	Mengembangkan spesifikasi kebutuhan <i>interface</i>	5	1	4	4	(3, 5, 6, 7)	23
22	Mengembangkan kebutuhan <i>software</i>	0	3	0	0	(4, 5, 6, 7)	23
23	<i>Review</i> spesifikasi setiap kebutuhan	5	1	1	1	(1, 2, 3, 4)	24
24	Mendesain tata letak awal	2	1	5	3	(8, 10, 11, 12)	25, 26, 27
25	Mendefinisikan <i>part-list</i> awal	2	1	5	3	(4, 5, 6, 7)	29
26	Mendefinisikan uji spesifikasi produksi awal	0	0	4	3	(4, 6, 7, 8)	28
27	Membuat dokumentasi (desain <i>software</i> , pengujian, dll)	0	4	0	2	(6, 7, 8, 9)	29
28	Membangun dan menguji sirkuit kritis <i>breadboard</i>	2	2	4	4	(10, 12, 13, 14)	29

No	Deskripsi Aktivitas	r <sub>1</sub>	r <sub>2</sub>	r <sub>3</sub>	r <sub>4</sub>	Durasi	Suksesor
29	Review rancangan desain	4	2	2	2	(1, 2, 3, 4)	30, 31, 32, 38, 39, 41
30	Membangun model <i>brassboard</i>	4	2	3	0	(9, 11, 12, 13)	33
31	Membuat model detail papan sirkuit	5	0	3	2	(10, 11, 13, 14)	36
32	Membuat model detail tata letak mesin	0	0	4	2	(8, 9, 10, 11)	37
33	Integrasi <i>software</i> fungsional	4	3	2	0	(6, 7, 8, 9)	34
34	Evaluasi fungsional	4	2	3	3	(3, 4, 5, 6)	35
35	Modifikasi inkorporat sirkuit kritis	2	2	0	4	(4, 5, 6, 7)	40
36	Menyusun kartu teknik model sirkuit	0	0	2	0	(1, 2, 3, 4)	40
37	Menyusun model <i>long-read-engineering</i> untuk bagian FAB	0	0	2	2	(3, 4, 5, 6)	40
38	Merencanakan kualifikasi <i>hardware</i> awal	1	0	3	1	(1, 2, 3, 4)	40
39	Menyusun bagian elektrik e-model beresiko rendah	0	0	2	2	(2, 3, 4, 5)	40
40	Review desain <i>hardware</i>	4	0	4	3	(1, 2, 3, 4)	44, 45, 48
41	Dokumentasi <i>software</i>	0	4	0	0	(3, 4, 5, 6)	42
42	Review dokumentasi kebutuhan <i>software</i>	3	3	0	0	(1, 2, 3, 4)	43
43	Review desain <i>software</i>	3	0	3	0	(1, 2, 3, 4)	44, 45
44	Pengujian <i>hardware</i> dan <i>software</i>	2	3	4	1	(3, 4, 5, 6)	51
45	Finalisasi dokumentasi	2	2	4	2	(4, 5, 6, 7)	46
46	Melengkapi data desain ke-	5	2	2	3	(4, 5, 6, 7)	47
47	Membuat BOM akhir	2	0	4	2	(3, 4, 5, 6)	51
48	Uji coba teknis	2	0	2	3	(5, 6, 7, 8)	49
49	Integrasi <i>software</i> dan <i>hardware</i>	4	2	3	2	(6, 7, 8, 10)	50
50	Review desain akhir	4	2	3	3	(2, 3, 4, 5)	51
51	Uji kuaifikasi	4	2	2	3	(2, 4, 6, 7)	-

Dari tabel dan keterangan suksesor di atas, maka dapat dibentuk diagram aktivitas untuk Proyek Pengembangan Elektronik yang ditunjukkan pada gambar (4) dan (5).

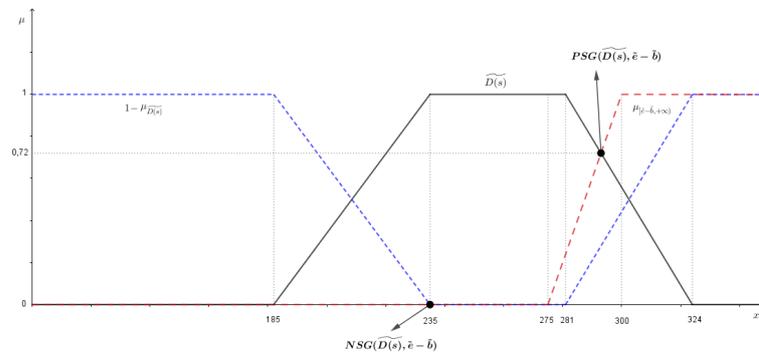


GAMBAR 4. Diagram aktivitas 1-23



GAMBAR 5. Diagram aktivitas 23-51

Dengan menerapkan metode RCPSP yang ditransformasikan menjadi barisan *Multi Knapsack Problem*, pembentukan jadwal ini akan menggunakan program *Python 3.8.8*. Jadwal yang terbentuk untuk proyek pengembangan produk elektronik di atas menghasilkan total durasi proyek  $\widetilde{D}(s) = (182, 235, 281, 324)$ ). Dengan *fuzzy ready time*  $\tilde{b} = (0, 0, 0, 0)$  dan *fuzzy deadline*  $\tilde{e} = (220, 245, 275, 300)$ . Selanjutnya jadwal tersebut kemudian dihitung nilai *schedule risk*-nya untuk mengetahui kemungkinan proyek yang dijalankan dengan jadwal tersebut akan selesai terlambat dari yang telah direncanakan di awal.



GAMBAR 6. Interpretasi  $PSG(\widetilde{D}(s), \tilde{e} - \tilde{b})$  dan  $NSG(\widetilde{D}(s), \tilde{e} - \tilde{b})$

Dari gambar 6, didapatkan nilai  $PSG(\widetilde{D}(s), \tilde{e} - \tilde{b})$  dan  $NSG(\widetilde{D}(s), \tilde{e} - \tilde{b})$  sebagai berikut

$$PSG(\widetilde{D}(s), \tilde{e} - \tilde{b}) = \sup_x \min(\mu_{\widetilde{D}(s)}(x), \mu_{|\tilde{e}-\tilde{b}, +\infty)}(x)) = 0.72,$$

$$NSG(\widetilde{D}(s), \tilde{e} - \tilde{b}) = \inf_x \max(1 - \mu_{\widetilde{D}(s)}(x), \mu_{|\tilde{e}-\tilde{b}, +\infty)}(x)) = 0.$$

Dengan menggunakan Persamaan (5.9) dan memilih nilai  $\beta = 0.5$ , maka *schedule-risk* untuk jadwal tersebut adalah:

$$\begin{aligned} SR(s) &= \beta \times PSG(\widetilde{D}(s), \tilde{e} - \tilde{b}) + (1 - \beta) \times NSG(\widetilde{D}(s), \tilde{e} - \tilde{b}) \\ &= (0.5 \times 0.72) + (1 - 0.5) \times 0 \\ &= 0.373. \end{aligned}$$

Sehingga didapatkan kemungkinan proyek yang dijalankan tersebut akan selesai terlambat dari yang telah direncanakan di awal sebesar 0.373.

## 7. KESIMPULAN

Permasalahan *Fuzzy* RCPSP dapat diselesaikan dengan menggunakan barisan Multi Knapsack. Pada simulasi kasus untuk penjadwalan proyek pengembangan produk elektronik, didapatkan solusi optimal dengan total durasi proyek sebesar (182,235,281,324) dan *schedule risk* sebesar 0,373. Hal tersebut menunjukkan bahwa dengan menggunakan Algoritma *Multi Knapsack*, maka proyek akan selesai pada (182,235,281,324) dengan kemungkinan jadwal tersebut akan selesai terlambat dari yang telah direncanakan di awal sebesar 0,373.

### Referensi

- [1] Ahuja, R.K., Magnanti, T.L., Orlin, J.B., *Network Flows*, Prentice Hall (1993), 71.
- [2] Bector, C.R., Chandra, S., *Fuzzy Mathematical Programming and Fuzzy Matrix Games*, Springer, New York, 2005.
- [3] Hillier, F.S., Lieberman, G.J., *Introduction to Optimization Research*, McGraw-Hill, pp 604-608, 2001.
- [4] Klir, G.J., Yuan, B., *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, Prentice-Hall, 1995.
- [5] Kolisch, R., Hartmann, S., Heuristical Algorithm for Solving The Resources-Constrained Project Scheduling Problem, *Classification and Computational Analysis*, Kluwer Academic Publisher(1999), 147-178.
- [6] Milos, S., *Solving Resources-Constrained Project Scheduling Problem As a Sequence of Multi-Knapsack Problems*, International Conference on Computers (2006), 80-86.
- [7] Pardalos, P.M., Aydogan, E.K., *Fuzzy Combinatorial Optimization Problems*, Handbook of Combinatorial Optimization (2013), 1357-1413.
- [8] Roudias, J., Mastering Principle and Practices in PMBOK, Prince 2, and Scrum, *Using Essential Project Management Methods to Deliver Effective and Efficient Projects*, Pearson Education (2015), 3-6.
- [9] Surinrat, C., Anussornnitisarn, P., Wongweragiat, S., International Journal of Emerging Technology and Innovative Engineering, *Application of Ranked Positional Weight in Detecting Resource Conflicts in Construction Project*, Vol.1, Issue 2 (2015)
- [10] Wang, J., *A Fuzzy Project Scheduling Approach to Minimize Schedule Risk for Product Development*, Fuzzy Set and System, Elsevier Science (2002), 99-116.
- [11] Wang, Y.M., Yang, J.B., Xu, D.L., Chin, K.S., *On the centroids of fuzzy numbers*, *Fuzzy Sets and Systems* (2006),919-926

SHIDKA HILDA MAGHFIROH

Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Sekip Utara Yogyakarta 55281, INDONESIA.

shidka.hilda.m@mail.ugm.ac.id

IRWAN ENDRAYANTO\* (Penulis Korespondensi)

Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Sekip Utara Yogyakarta 55281, INDONESIA.

endrayanto@ugm.ac.id