

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Persaingan di era global telah membuat banyak perusahaan ikut berkembang menjadi perusahaan berskala global, hal tersebut ditandai dengan makin banyak perusahaan yang memiliki rantai pasok yang tersebar dipenjuruk dunia (Chopra dan Meindl, 2013). Tidak jarang sebuah perusahaan global memiliki pabrik produksi di satu benua, namun memiliki gudang penyimpanan di benua lainnya. Hal tersebut dilakukan untuk memudahkan pemasaran produk keseluruh dunia dan menekan biaya produksi. Pengelolaan rantai pasok sangat dipengaruhi oleh pemilihan lokasi dan kualitas barang sehingga diperlukan pemilihan yang tepat agar dapat memberi keuntungan yang maksimal bagi perusahaan (Heizer dan Render, 2011). Salah satu pemilihan yang sering terjadi pada rantai pasok adalah pemilihan *supplier* pemasok bahan baku. Pemilihan *supplier* dapat mempengaruhi kualitas barang yang dihasilkan bahkan mempengaruhi performa dari rantai pasok organisasi yang ada (Chen dkk., 2006; Thruilogachantar dan Zailani, 2011). Melihat pentingnya efek yang dihasilkan dari pemilihan supplier ini, maka pemilihan supplier digunakan penulis sebagai topik utama dalam penelitian.

Proses pemilihan supplier tentu bukanlah hal yang sederhana, banyak sekali faktor yang harus diperhitungkan dalam proses pemilihannya agar solusi yang dihasilkan dapat memberikan manfaat maksimal bagi perusahaan. Faktor pemilihan ditentukan atas infrastruktur yang digunakan, preferensi pengambil kebijakan, serta konstrain-konstrain lainnya yang bersifat universal (Owen dan Daskin, 1998; Farahani dan Hekmatfar, 2009). Melihat tingkat kompleksitas yang diperlukan dalam pemilihan supplier tersebut, maka diperlukan sebuah metode yang dapat membantu penulis dalam memberikan solusi terbaik bagi penelitian ini.

Salah satu metode yang tepat digunakan untuk pemilihan dengan banyak alternatif adalah *Multi Criteria Decision Making* (MCDM) (Creed, 1968). MCDM adalah metode yang digunakan untuk memilih solusi dari beberapa alternatif solusi yang ada berdasarkan variabel-variabel yang menjadi konstrainnya (Rao, 2013).

MCDM menggunakan prosedur matematika dan perhitungan komputasi untuk menentukan solusi dari masalah yang memiliki banyak konstrain (Figueira dkk., 2005). Perhitungan MCDM memiliki langkah-langkah yang dapat digunakan untuk menentukan alternatif dari beberapa alternatif yang ada (Majumder, 2015). Salah satu contoh metode MCDM adalah AHP.

Salah satu aplikasinya adalah penggunaan *fuzzy* AHP dan VIKOR untuk menentukan suplai global yang *sustainable*, peneliti mencoba untuk menggabungkan metode *fuzzy* AHP dan VIKOR untuk menentukan lokasi supplier terbaik secara global yang *sustainable* dengan kriteria yang berskala global juga seperti: faktor ekonomi, faktor kualitas, faktor lingkungan, faktor resiko global, serta faktor social. Hasil penelitian menyimpulkan bahwa kedua metode tersebut mampu menentukan supplier terbaik berskala global walaupun dengan jumlah informasi kuantitatif yang sedikit, selain itu diketahui bahwa diantara 5 faktor global yang ada, diketahui bahwa faktor ekonomi tetap memiliki pengaruh paling besar terhadap pemilihan supplier (Awasthi dkk., 2017). Melihat hasil penelitian tersebut, maka faktor ekonomi atau yang berhubungan langsung dengan keuangan dipertimbangkan penulis dalam kriteria sistem.

Penggunaan *Fuzzy* AHP dan *Fuzzy* TOPSIS untuk menentukan suplai terbaik, dari hasil penelitian diketahui bahwa terdapat 6 kriteria utama yang diperhatikan dalam menentukan sebuah metode dalam pemilihan lokasi yaitu: fleksibilitas terhadap perubahan alternatif maupun kriteria, kelincahan dalam pengambilan keputusan, tingkat kompleksitas dalam komputasi, dukungan untuk model *Group decision making*, jumlah kriteria dan alternatif yang didukung, serta kemampuan memodelkan ketidakpastian dalam perhitungan. *Fuzzy* AHP dan *fuzzy* TOPSIS diketahui lulus terhadap 6 kriteria tersebut. Selain itu, diketahui juga bahwa *Fuzzy* AHP baik digunakan ketika jumlah informasi sedikit dan kurang lengkap (Junior dkk., 2014). Melihat hasil penelitian tersebut, penulis dapat menggunakan *fuzzy* AHP sebagai salah satu metode yang tepat digunakan dalam penelitian ini karena tidak perlu membutuhkan banyak informasi untuk mendapatkan hasil pemilihan supplier yang diinginkan.

Selanjutnya penggunaan model transportasi *linier programming* untuk menentukan pemilihan lokasi gudang terbaik. Penelitian menambahkan salah satu faktor dalam menentukan lokasi gudang yaitu biaya transportasi. Dari hasil penelitian diketahui bahwa biaya transportasi memberikan pengaruh yang sangat besar dalam memilih lokasi gudang dengan menggunakan metode linier programming (Khan dan Zaidi, 2013). Melihat hasil penelitian tersebut, maka faktor biaya transportasi dipertimbangkan penulis ke dalam alternatif pemilihan supplier.

Selanjutnya adalah faktor *tangible* dan *intangible* dalam menentukan suatu pemilihan. Dalam jurnal Khan dan Zaidi, (2013) diketahui bahwa linier programming hanya berfokus pada faktor *tangible* saja, padahal faktor *intangible* yang tidak memberikan dampak langsung seperti insentif dan preferensi ternyata ikut memberikan pengaruh yang signifikan (Ghodsypour dan O'Brien, 1998). Pada metode AHP diketahui bahwa metode AHP dapat menggunakan faktor *tangible* dan *intangible* sekaligus. Namun, input data AHP hanya didasarkan pada pemilihan derajat kepentingan dimana hal tersebut dapat memberikan subjektifitas yang tinggi, sebaliknya metode MOLP menggunakan data history di lapangan sehingga tingkat subjektifitasnya sangat rendah. Maka, metode AHP dan metode MOLP memiliki pendekatan yang berbeda.

Melihat tinjauan beberapa jurnal yang sudah dibahas sebelumnya, maka menggabungkan metode AHP dengan MOLP yang disertai dengan *fuzzy* dapat digunakan untuk menyelesaikan permasalahan pemilihan supplier. Penelitian yang mencoba menggabungkan metode AHP maupun MOLP dengan metode lain juga telah banyak dilakukan. Pertama, penggunaan metode *linier programming* dan TOPSIS untuk menentukan masalah banyak atribut yang bersifat *fuzzy* (Zhang dkk., 2015). Dari hasil penelitian tersebut diketahui bahwa *fuzzy* dapat membantu mengurangi ketidakpastian dalam perhitungan masalah banyak atribut. Selain itu penggunaan *fuzzy* diperlukan untuk pendamping metode MCDM seperti AHP dalam pemilihan dengan solusi yang mengandalkan model *trade-off* seperti pemilihan *Supplier* (Kumar dan Singh, 2012). Berdasarkan hasil penelitian tersebut,

maka penulis menggunakan *fuzzy sets* ke dalam metode yang digunakan dalam penelitian.

Kedua, penelitian tentang penggunaan *Interval-Valued Intuitionistic Fuzzy Sets*, metode *Linear Programming*, dan *Extended TOPSIS Method* untuk menentukan masalah banyak atribut (Wang dan Chen, 2017). Dari penelitian tersebut, diketahui bahwa Linier programming dapat digabungkan dengan metode lain untuk mendapatkan solusi yang lebih baik, maka penggabungan metode linier programming dengan metode lain sangatlah dimungkinkan. Ketiga, penggunaan metode *linier programming* dan *fuzzy AHP* untuk menentukan lokasi pemilihan fasilitas dalam jumlah banyak (Ozgen dan Gulsun, 2014). Dari studi literatur, penggabungan metode MCDM dan Model Transportasi dapat dilakukan untuk optimasi penentuan *supplier* terbaik.

Metode *Fuzzy MOLP* dan *Fuzzy AHP* tidak selalu memiliki hasil perbandingan yang sama, maka untuk mengantisipasi hal tersebut, diperlukan sebuah pengujian. Pengujian yang dilakukan adalah analisis sensitivitas. Analisis sensitivitas dapat digunakan untuk menguji ketidaksensitifan suatu metode. Metode yang tidak sensitif merupakan metode yang lebih baik. Analisis sensitifitas pernah dilakukan pada pengujian dua metode yang berbeda. Dalam penelitian Indriyati dkk., (2013), analisis sensitivitas dilakukan pada metode AHP dan TOPSIS untuk mengetahui metode mana yang lebih tidak sensitif. Dari hasil penelitian diketahui bahwa metode AHP jauh lebih tidak sensitif terhadap perubahan dibandingkan dengan metode TOPSIS. Berdasarkan penelitian tersebut, maka penulis menggunakan analisis sensitifitas untuk menguji ketidaksensitifan antara metode *fuzzy AHP* dan *fuzzy MOLP*. Namun, perlu dilakukan analisis apakah metode yang tidak sensitif akan selalu lebih baik dari metode yang lebih sensitif, terutama metode AHP yang dihipotesiskan akan lebih tidak sensitif daripada linier programming.

2.2. Dasar Teori

2.2.1. Pemilihan *Supplier*

Pemilihan *supplier* merupakan salah satu hal paling penting bagi dunia bisnis. Biaya transportasi sendiri dapat memakan hingga 25% harga jual barang, maka

pemilihan yang tepat dapat memberikan keuntungan maksimal bagi perusahaan (Heizer dan Render, 2011). Pemilihan selalu identik dengan jarak, waktu, serta klien atau tujuan yang ingin dicapai (Farahani dan Hetmakfar, 2009). Biasanya pemilihan dilakukan berdasarkan jarak terdekat dengan konsumen atau terdekat dengan fasilitas tertentu diperusahaan (Farahani dan Hetmakfar, 2009). Pemilihan *supplier* memiliki beberapa faktor yang harus diperhatikan sebelum memilih lokasi bagi fasilitas tersebut. Terdapat empat faktor yang harus diperhatikan oleh perusahaan dalam pemilihan *supplier*-nya :

- a) Kapasitas (Ozgen dan Gulsun, 2014; Khan dan Zaidi, 2013; Ghodsypour dan O'Brien, 1998)

Kapasitas menggambarkan berapa produksi yang dapat dihasilkan oleh *supplier* dalam jangka waktu tertentu. Selain kapasitas produksi *supplier*, kapasitas pengangkutan transportasi juga menjadi hal yang penting karena truk pengangkut dengan kapasitas besar dapat membawa suplai lebih banyak namun memerlukan biaya dan waktu relatif lebih banyak juga. Maka, pemilihan tidak hanya memilih kapasitas produksi dan pengangkutan yang paling besar namun kapasitas produksi dan pengangkutan yang paling tepat.

- b) Waktu (Farahani dan Hetmakfar, 2009; Kumar dan Singh, 2012; Ghodsypour dan O'Brien, 1998)

Waktu merupakan hal yang paling utama dalam transportasi sehingga keterlambatan pengiriman dapat menyebabkan kerugian yang besar bagi sebuah perusahaan seperti berhentinya produksi akibat bahan baku produksi mengalami keterlambatan. Mempercepat waktu pengiriman harus dilakukan oleh perusahaan agar aktivitas produksi perusahaan tidak terganggu.

- c) Biaya (Heizer dan Render, 2011; Kumar dan Singh, 2012; Khan dan Zaidi, 2013; Ghodsypour dan O'Brien, 1998)

Dalam faktor biaya, terdapat dua jenis biaya yang harus diperhatikan, yaitu : *tangible cost* dan *intangible cost*. *Tangible cost* adalah biaya yang terukur dan jelas seperti gaji, material, pajak dan lain-lain. *Intangible cost* adalah biaya yang tidak dapat terukur seperti pandangan masyarakat, kebijakan pemerintah, dan lain-lain. Selain *tangible cost*, *intangible cost* adalah biaya yang perlu

diperhatikan oleh perusahaan, terkadang pengabaian *intangible cost* dapat menyebabkan efek yang permanen dan berkepanjangan bagi perusahaan.

d) Jarak (Farahani dan Hetmakfar, 2009)

Kedekatan dengan pabrik produksi juga sangat penting dengan pemilihan *supplier*. Kedekatan antara pabrik produksi dengan suplai bahan baku akan memudahkan dalam pendistribusian bahan baku terutama untuk perusahaan yang menerapkan konsep *Just-in-time* dimana keterlambatan bahan baku dapat mengakibatkan berhentinya proses produksi.

2.2.2. Model Transportasi

Model transportasi adalah salah satu metode yang sering digunakan dalam menentukan pemilihan *supplier*. Model transportasi adalah metode yang digunakan untuk menyelesaikan sebuah permasalahan progama linier (Heizer dan Render, 2011). Tujuan utama dari model transportasi adalah menemukan jalur pendistribusian terbaik dari beberapa titik suplai bahan baku menuju beberapa titik tujuan pengiriman dengan biaya pengiriman dan produksi terendah (Heizer dan Render, 2011). Model transportasi banyak digunakan untuk menyelesaikan permasalahan di dunia nyata. Salah satu contoh model transportasi adalah *Linier Programming*.

2.2.3. Fuzzy Multi Objective Linier Programming (F-MOLP)

Dalam model transportasi, terdapat empat masalah utama yang sering menjadi target perbaikan dalam model transportasi, yaitu: *Customer service*, lokasi fasilitas pabrik, keputusan inventori, dan keputusan transportasi (Khan dan Zaidi, 2013). *Linier programming* (LP) merupakan bagian dari model transportasi. LP adalah metode matematis yang biasa digunakan oleh manajer untuk membuat keputusan yang biasa memerlukan *trade-off* dalam pengalokasian sumber daya (Heizer dan Render, 2011). LP sendiri sudah sering sekali digunakan dalam menyelesaikan permasalahan terutama pada pemilihan *supplier*. Dalam jurnalnya digunakan metode LP untuk menyelesaikan permasalahan pemilihan lokasi pergudangan (Khan dan Zaidi, 2013). Untuk jenis LP yang digunakan dalam penelitian ini, jenis yang digunakan adalah *Multiple Objective Linier Programming*

(MOLP). MOLP digunakan karena kasus yang digunakan adalah pemilihan supplier bahan baku terbaik dari banyak pilihan supplier yang ada.

Pada pemilihan lokasi, hasil dari MOLP dinilai kurang optimal karena solusi pemilihan lokasi yang terkadang tidak memberikan hasil yang pasti sehingga penggunaan *fuzzy* diperlukan untuk pendamping metode MCDM seperti MOLP dalam pemilihan dengan solusi yang mengandalkan model *trade-off* seperti pemilihan tempat (Kumar dan Singh, 2012). Maka, penambahan *fuzzy* pada MOLP diperlukan dalam penelitian ini.

Langkah-langkah dalam *Fuzzy* MOLP antara lain:

1. Membuat model MOLP untuk permasalahan

Model MOLP yang digunakan telah dimodifikasi dengan nilai *fuzzy*, sehingga dengan memodifikasi persamaan (2.1) akan didapatkan persamaan :

$$\begin{array}{ll} \text{Maksimasi} & f(x) = C_x \\ \text{Subjek} & x \in X = \{x \in R^n | \tilde{A}x \leq \tilde{b}, x \geq 0\} \end{array} \quad (2.1)$$

dimana:

C_x = matriks $k \times n$ dengan tiap elemen diubah menjadi bentuk *fuzzy*

A = matriks $m \times n$ dengan tiap elemen diubah menjadi bentuk *fuzzy*.

b = vektor m

x = vektor n dari variabel pengambilan keputusan

2. Mengubah model MOLP menjadi *Fuzzy* MOLP

Setelah model permasalahan telah terbentuk, maka konversi model MOLP menjadi *fuzzy* MOLP, sehingga menjadi persamaan dibawah:

$$\begin{array}{ll} \text{Maksimasi} & \left\{ \frac{C_{\lambda}^L x}{C_{\lambda}^R x} \right\}, \forall \lambda \in [0,1] \\ \text{Subjek} & x \in X = \{x \in R^n | A_{\lambda}^L x \leq b_{\lambda}^L x, A_{\lambda}^R x \leq b_{\lambda}^R, x \geq 0, \forall \lambda \in [0,1]\} \end{array} \quad (2.2)$$

dimana:

$$C_{\lambda}^L = \begin{bmatrix} c_{11\lambda}^L & c_{12\lambda}^L & \cdots & c_{1n\lambda}^L \\ c_{21\lambda}^L & c_{22\lambda}^L & \cdots & c_{2n\lambda}^L \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1\lambda}^L & c_{k2\lambda}^L & \cdots & c_{kn\lambda}^L \end{bmatrix} \quad C_{\lambda}^R = \begin{bmatrix} c_{11\lambda}^R & c_{12\lambda}^R & \cdots & c_{1n\lambda}^R \\ c_{21\lambda}^R & c_{22\lambda}^R & \cdots & c_{2n\lambda}^R \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1\lambda}^R & c_{k2\lambda}^R & \cdots & c_{kn\lambda}^R \end{bmatrix}$$

$$A_{\lambda}^L = \begin{bmatrix} a_{11\lambda}^L & a_{12\lambda}^L & \cdots & a_{1n\lambda}^L \\ a_{21\lambda}^L & a_{22\lambda}^L & \cdots & a_{2n\lambda}^L \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1\lambda}^L & a_{k2\lambda}^L & \cdots & a_{kn\lambda}^L \end{bmatrix} A_{\lambda}^R = \begin{bmatrix} a_{11\lambda}^R & a_{12\lambda}^R & \cdots & a_{1n\lambda}^R \\ a_{21\lambda}^R & a_{22\lambda}^R & \cdots & a_{2n\lambda}^R \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1\lambda}^R & a_{k2\lambda}^R & \cdots & a_{kn\lambda}^R \end{bmatrix}$$

$$b_{\lambda}^L = [b_{1\lambda}^L, b_{2\lambda}^L, \dots, b_{m\lambda}^L]^T \quad b_{\lambda}^R = [b_{1\lambda}^R, b_{2\lambda}^R, \dots, b_{m\lambda}^R]^T$$

C_x = matriks k x n dengan tiap elemen diubah menjadi bentuk *fuzzy*

A = matriks m x n dengan tiap elemen diubah menjadi bentuk *fuzzy*.

b = vektor m

x = vektor n dari variabel pengambilan keputusan

3. Perhitungan solusi dengan simpleks

Setelah model *Fuzzy* MOLP telah ditemukan, maka perhitungan dapat dilakukan dengan perhitungan simpleks normal atau simpleks biasa. Perhitungan simpleks adalah sebuah prosedur perhitungan aljabar yang memiliki sifat iteratif, yaitu proses penyelesaian dilakukan secara bertahap dari titik *infeasible* (belum optimum) hingga mencapai titik *feasible* (optimum) (Dimyati dan Dimyati, 2013). Perhitungan simpleks sama seperti perhitungan LP biasa, namun penyelesaian solusi dilakukan tahap pertahap hingga ditemukan satu-satunya solusi yang dapat dianggap optimum. Perhitungan simpleks memudahkan perhitungan LP namun membutuhkan proses perhitungan lebih panjang. Hasil dari perhitungan simpleks merupakan solusi terbaik.

4. Proses *defuzzy* nilai *fuzzy*

Hasil dari perhitungan *fuzzy* adalah angka *fuzzy*, maka diperlukan perubahan nilai *fuzzy* menjadi angka nyata sehingga dapat dilakukan perbandingan. Proses konversi dilakukan dengan nilai *Best Nonfuzzy Performance* (BNP) dengan metode *centre-of-area* dengan persamaan :

$$BNP_j = \frac{[(u_i - l_i) + (m_i - l_i)]}{3 + l_i}$$

(2.3)

dimana:

u = nilai triangular *fuzzy* pertama

m = nilai triangular *fuzzy* kedua

l = nilai triangular *fuzzy* ketiga

i = nilai ke-i dari nilai *fuzzy* yang digunakan

j = nilai ke-j dari proses de-*fuzzy*-fikasi yang dilakukan

2.2.4. Multi Criteria Desicion Making (MCDM)

Pemilihan fasilitas perusahaan merupakan perhitungan yang kompleks (Heizer dan Render, 2011). Maka, sebuah metode dalam membantu mengambil keputusan sangatlah diperlukan. Salah satu metode yang biasa digunakan untuk memilih lokasi fasilitas adalah model MCDM. MCDM adalah metode yang digunakan untuk memilih solusi dari beberapa alternatif solusi yang ada berdasarkan variabel-variabel yang menjadi konstrainnya (Rao, 2013). Dari tabel 2.1, dapat dilihat bahwa MCDM memilih solusi terbaik berdasarkan atribut atau variabel yang ada. Alternatif solusi yang ada tergantung jumlah alternatif yang ada, sedangkan untuk jumlah atributnya tergantung jumlah konstrain yang dibutuhkan. Dalam melakukan perhitungan MCDM, terdapat beberapa langkah yang harus dilakukan (Majumder, 2015).

Tabel 2.1 Contoh model penyelesaian MCDM

Alternatif	Atribut					
	B ₁ (W ₁)	B ₂ (W ₂)	B ₃ (W ₃)	- (-)	- (-)	B _M (W _M)
A ₁	m ₁₁	m ₁₂	m ₁₃	-	-	m _{1M}
A ₂	m ₂₁	m ₂₂	m ₂₃	-	-	m _{2M}
A ₃	m ₃₁	m ₃₂	m ₃₃	-	-	m _{3M}
-				-	-	
-				-	-	
A _N	m _{N1}	m _{N2}	m _{N3}			m _{NM}

dimana:

A = alternatif

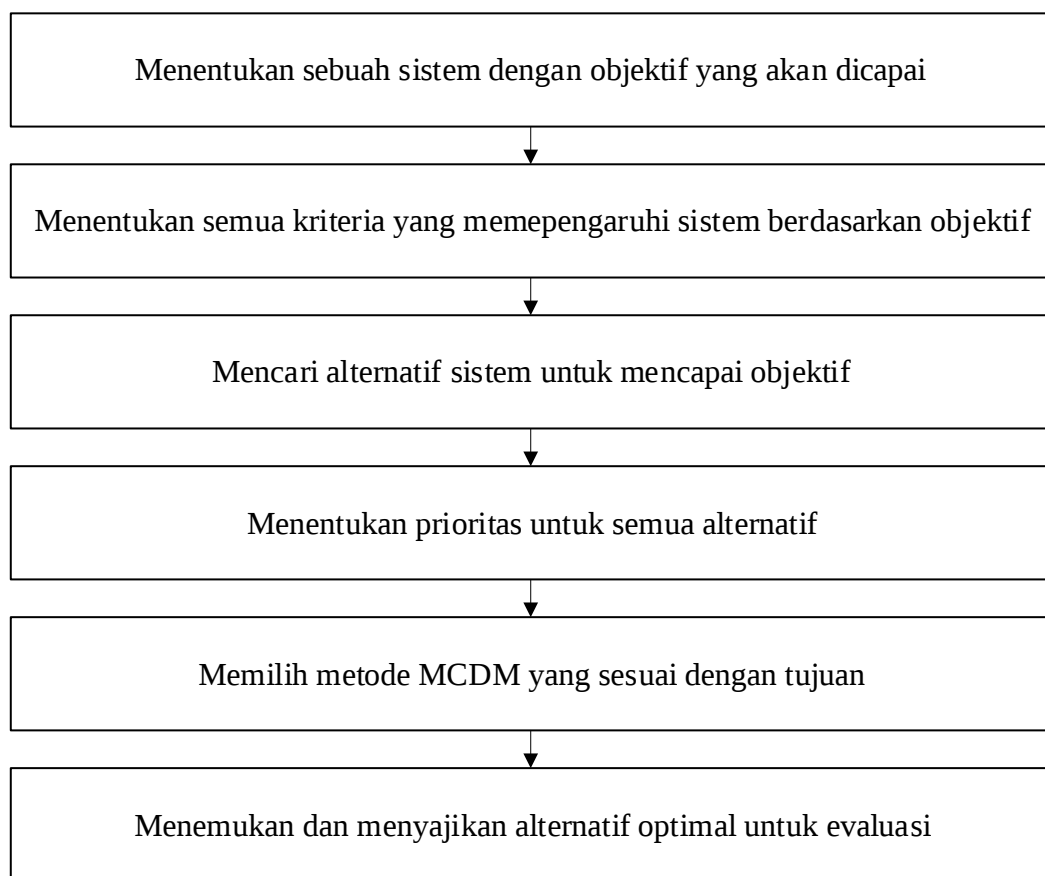
B = Kriteria

W = jenis kriteria

N = baris Matriks

M = kolom matriks

m = matriks ke NM



Gambar 2.1 Alur Proses MCDM (Majumder, 2015)

Dari Gambar 2.1, diperlihatkan bahwa MCDM memilih beberapa langkah yang dilakukan untuk melakukan perhitungan, yaitu :

1. Identifikasikan tujuan atau goal dalam proses pengambilan keputusan
Tujuan menjadi akhir dari perhitungan sehingga didapatkan hasilnya.
2. Pemilihan kriteria/atribut/parameter sebagai konstanta

Kriteria yang dipilih haruslah : berkoherensi dengan keputusan, independen antar satu dan lainnya, memiliki skala yang sama, terukur, dan tidak berhubungan dengan alternatif solusi yang akan dipilih.

3. Pemilihan alternatif solusi

Alternatif solusi yang ada haruslah : tersedia, dapat dibandingkan, nyata bukan hanya ideal, praktikal dan dapat diterapkan.

4. Pemilihan faktor pemberat sebagai skala derajat kepentingan

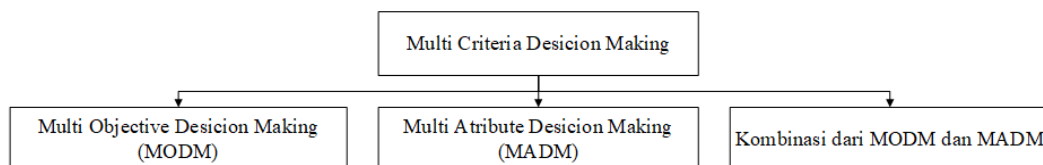
Faktor pemberat adalah faktor yang digunakan untuk melakukan perangkingan alternatif solusi yang ada. Faktor pemberat haruslah dapat membuat rangking atau dapat membuat pembandingan.

5. Pemilihan metode agregasi

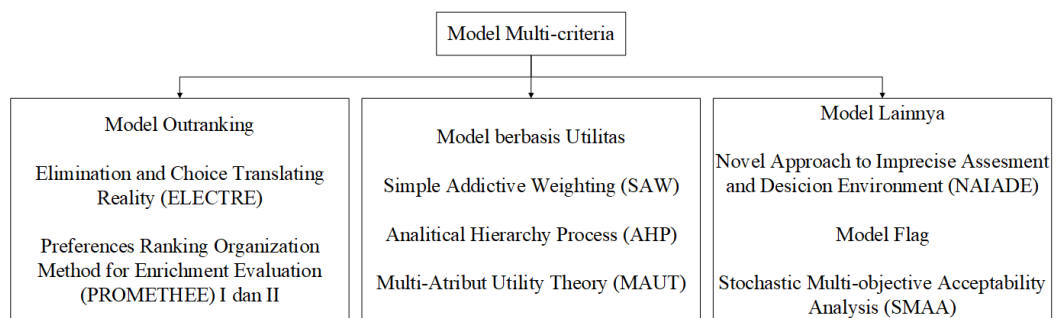
Metode agregasi haruslah : berupa produk, rata-rata, fungsi, atau sesuatu hal yang dapat memisahkan atau memilih satu alternatif solusi dari alternatif solusi lainnya.

6. Pengambilan keputusan berdasarkan hasil agregasi

Setelah hasil agregasi didapatkan, maka keputusan solusi terbaik telah ditemukan dan dapat digunakan.



Gambar 2.2 Turunan Dari MCDM (Mateo, 2012)



Gambar 2.3 Jenis-Jenis Model MCDM (Polatidis.dkk, 2006)

MCDM sendiri merupakan metode yang masih sangat luas sekali, sehingga banyak metode yang menjadi turunan MCDM. MCDM terbagi menjadi tiga, yaitu : *Multi Obejctive Decision Making* (MODM), *Multi Atribute Desicion Making* (MADM), dan kombinasi antara MODM dan MADM (Mateo, 2012). Turunan yang ada hanya berbeda pada variabel atau konstrain yang digunakan. Bagan turunan MCDM dapat dilihat pada Gambar 2.2. Selain memiliki turunan, model perhitungan dari MCDM juga terbagi menjadi tiga (Polatidis dkk., 2006) yang dapat dilihat pada Gambar 2.3. Gambar 2.3 memperlihatkan tiga pembagian model MCDM, yaitu :

a) *Outranking Models*

Model ini menggunakan perankingan pada tiap alternatif solusi yang ada berdasarkan atribut atau variabelnya yang paling penting, alternatif yang di bawah ranking yang terbaik akan dieliminasi yang artinya variabel tersebut tidak akan dihitung pada variabel lainnya (Majumder, 2015). Metode ini sangat baik bila terdapat variabel yang mendominasi atau memiliki derajat kepentingan diatas variabel lainnya, sehingga ketika alternatif terpilih memiliki *score* buruk divariabel lainnya, hal tersebut tidak terlalu mempengaruhi hasil karena variabel tersebut memiliki “*impact*” yang tidak terlalu besar dibandingkan variabel utama. Model ini sangat baik digunakan ketika terdapat faktor keberpihakan atau politis dalam pemilihan, dimana hal tersebut adalah hal yang biasa di dunia nyata (Majumder, 2015). Kelemahan model ini adalah bila tidak terdapat kriteria yang mendominasi dari kriteria lainnya sehingga tidak dapat dipilih alternatif dengan dominasi pada kriteria pilihan.

Contoh dari *Outranking models* adalah PROMETHEE dan ELECTREE.

b) *Utility-based Models*

Utility-based atau *compensatory models* menggunakan berbagai kriteria atau atribut yang ada dalam memilih alternatif solusi terbaik (Majumder, 2015). Alternatif yang memiliki *score* yang baik di satu variabel dapat membantu pada variabel yang memiliki *score* buruk sehingga semua variabel memiliki derajat kepentingan yang sama dan hasil yang didapatkan adalah alternatif dengan nilai keseluruhan terbaik. Metode ini sangat baik bila semua kriteria memiliki derajat kepentingan yang sama sehingga hasil yang didapatkan biasanya berbentuk

trade-off (Majumder, 2015). Kelemahan dari model ini adalah bila kriteria merupakan variabel yang tidak dapat dikompensasi, sehingga *trade-off* tidak mungkin dilakukan.

Contoh dari *Utility-based models* adalah AHP, SAW, dan MAUT

c) *Miscellaneous Models*

Miscellaneous models adalah model yang tidak mengikuti dua aturan sebelumnya, artinya tiap model atau metode memiliki cara perhitungan sendiri-sendiri. Kelebihan model ini adalah berguna untuk beberapa kasus unik yang tidak dapat ditangani oleh dua model sebelumnya. Namun, kelemahan model-model ini adalah tidak banyak digunakan serta sulitnya dikombinasikan dengan dua model sebelumnya karena perbedaan secara fundamental.

Contoh dari *miscellaneous models* adalah NAIADE, flag model, dan SMAA

2.2.5. Fuzzy Analytical Hierarchy Process (F-AHP)

Model MCDM yang digunakan dalam penelitian ini adalah *Analytical Hierarchy Process* (AHP). Metode AHP dipilih dalam penelitian ini karena elemen yang digunakan dalam pemilihan *supplier* terdiri dari banyak atribut dan sub-atribut sehingga diperlukan proses pemecahan masalah menjadi lebih sederhana agar lebih mudah diselesaikan, maka metode AHP yang memiliki cara kerja membagi suatu permasalahan ke dalam bentuk hierarki sangatlah tepat digunakan dalam penelitian ini. AHP adalah salah satu model *compensatory* dimana AHP membantu pengambilan keputusan dengan memilih solusi yang paling mendekati tujuan dan menyelesaikan permasalahan, bukan solusi yang paling baik diantara lainnya (Majumder, 2015). AHP membagi permasalahan dalam bentuk hierarki dan sub masalah sehingga permasalahan akan mudah dilihat dari struktur umum hingga mendetail. Dengan begitu permasalahan akan lebih mudah dianalisis dan ditemukan solusinya daripada menyelesaikan permasalahan secara keseluruhan.

Pada pemilihan lokasi, metode MCDM seperti AHP dinilai kurang optimal karena solusi pemilihan lokasi yang tidak pasti. Penggunaan *fuzzy* diperlukan untuk pendamping metode MCDM dalam pemilihan dengan solusi yang mengandalkan

model *trade-off* seperti pemilihan tempat (Kumar dan Singh, 2012). Maka, penambahan *fuzzy* pada AHP diperlukan dalam penelitian ini.

Langkah-langkah dalam *Fuzzy AHP* antara lain (Ozdemir dan Ayag, 2006):

1. Menentukan pemilihan kriteria

Tabel 2.2 Skala Nomer *Pairwise comparison*

Derajat Kepentingan	Keterangan
1	Kedua kriteria memiliki kepentingan yang sama
3	Salah satu kriteria sedikit lebih penting dari kriteria lainnya
5	Salah satu kriteria lebih penting dari kriteria lainnya
7	Salah satu kriteria sangat lebih penting dari kriteria lainnya
9	Salah satu kriteria mutlak lebih penting dari kriteria lainnya

Kemudian matriks diisi dengan angka derajat kepentingan yang dapat dilihat pada tabel 2.2. Dengan menjumlahkan nilai-nilai dari tiap kolom pada matriks, membagi setiap nilai dari kolom dengan kolom total, kemudian menjumlahkan nilai dari setiap matriks dan membaginya dengan jumlah kriteria akan mendapatkan nilai rata-rata. Namun, karena bentuk matriks adalah triangular, maka perlu dilakukan konversi nilai pada hasil pemilihan nilai *pairwise comparison*.

Berikut ini nilai konversi yang dilakukan:

$$1_{\alpha} = [1, 3 - 2\alpha], \quad \sim \quad (2.4)$$

$$3_{\alpha} = [1 + 2\alpha, 5 - 2\alpha], \quad 3_{\alpha}^{-1} = \left[\frac{1}{5 - 2\alpha}, \frac{1}{1 + 2\alpha} \right]$$

$$5_{\alpha} = [3 + 2\alpha, 7 - 2\alpha], \quad 5_{\alpha}^{-1} = \left[\frac{1}{7 - 2\alpha}, \frac{1}{3 + 2\alpha} \right]$$

$$7_{\alpha} = [5 + 2\alpha, 9 - 2\alpha], \quad 7_{\alpha}^{-1} = \left[\frac{1}{9 - 2\alpha}, \frac{1}{5 + 2\alpha} \right]$$

$$9_{\alpha} = [7 + 2\alpha, 11 - 2\alpha], \quad 9_{\alpha}^{-1} = \left[\frac{1}{11 - 2\alpha}, \frac{1}{7 + 2\alpha} \right]$$

Dengan α adalah koefisien alpha dengan nilai $0 < \alpha < 1$

2. Menentukan *eigen vector* dari tiap elemen

Eigen vector dapat ditemukan dengan menggunakan persamaan :

$$V_i = (\bar{a}_{11} * \bar{a}_{12} * \bar{a}_{13} * \dots * \bar{a}_{1n})^{1/n}$$

(2.5)

3. Normalisasi *eigen vector*

Proses normalisasi dilakukan dengan menggunakan persamaan :

$$w_i = \left(\frac{v_l}{1/\sum v_l}, \frac{v_m}{1/\sum v_m}, \frac{v_u}{1/\sum v_u} \right)$$

(2.6)

dan

$$T = \left(w_1/\sum w_i, w_2/\sum w_i, w_3/\sum w_i, \dots, w_n/\sum w_i \right)$$

(2.7)

4. Proses *defuzzy* nilai *fuzzy*

Hasil dari perhitungan *fuzzy* adalah angka *fuzzy*, maka diperlukan perubahan nilai *fuzzy* menjadi angka nyata sehingga dapat dilakukan perangkingan. Proses konversi dilakukan dengan nilai *Best Nonfuzzy Performance* (BNP) dengan metode *centre-of-area* dengan persamaan (2.3).

5. Melakukan uji konsistensi

Pengukuran rasio konsistensi dilakukan untuk mendapatkan hasil yang dapat merepresentasikan dunia nyata. Rasio konsistensi (CR) memiliki nilai yang berbeda sesuai dengan ukuran matriksnya. CR didapatkan dengan persamaan :

$$\lambda_{max} = \frac{1}{n} \sum_{i=1}^n \left(\frac{\text{Kriteria ke } i \text{ pada } A}{\text{Kriteria Ke } i \text{ pada } W^t} \right) \quad (2.8)$$

Setelah λ_{max} didapatkan, maka nilai CI dapat ditemukan dengan persamaan :

$$CI = \frac{(\lambda_{max} - n)}{n - 1} \quad (2.9)$$

Dimana :

CI = *Consistency Index*

n = banyak kriteria

Setelah CI didapatkan, kemudian CR dapat ditemukan dengan menggunakan persamaan :

$$CR = \frac{CI}{RC} \quad (2.10)$$

Dimana :

CR = *Consistensy Ratio*

CI= *Consistensy Index*

RC= *Random consistensy*

Nilai RC dapat dilihat pada tabel 2.3

Tabel 2.3 tabel nilai random consistensy (RC) (Saaty, 1980)

n	1	2	3	4	5	6	7	8	9
RC	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

6. Bila nilai $CR < 0.1$ berarti perhitungan dapat diterima sehingga dapat dilanjutkan.
7. Hasil *defuzzy* dari perhitungan nomer 5 akan menghasilkan *performance value*. Alternatif dengan *performance value* tertinggi akan menjadi prioritas utama, dilanjutkan dengan alternatif dengan *performance value* tertinggi kedua dan seterusnya.
8. Alternatif dengan Prioritas utama menjadi solusi terbaik diikuti dengan alternatif dengan prioritas kedua dan seterusnya.

2.2.6. *Rational Unified Process (RUP)*

Rational Unified Process (RUP) adalah salah satu metode dalam *Software Engineering* yang memiliki pendekatan yang ketat dalam pembagian tugas dan tanggung jawab di tiap proses pengembangan (Kruchten, 1999). Hal tersebut bertujuan agar *Software* yang dihasilkan memiliki kualitas tinggi yang dapat memenuhi kebutuhan *end-user*, namun tetap dapat diperoleh dalam jangka waktu dan *budget* yang telah ditentukan di awal. RUP merupakan metode yang bersifat iteratif dimana proses pengembangannya dilakukan berulang-ulang hingga *Software* yang dihasilkan sesuai dengan tujuan awal. Selain bersifat iteratif, RUP juga lebih berfokus pada pengembangan model atau arsitektur, sehingga RUP dapat memanfaatkan bahasa *Unified Modelling Language (UML)*. UML adalah bahasa permodelan yang menjadi standar pada bidang industri, dengan menggunakan

UML maka penggambaran kebutuhan, desain, dan model *Software* akan jauh lebih mudah dilakukan terutama dengan klien yang menjadi *end-user* dari *Software*. Walaupun memiliki aturan yang ketat, RUP memungkinkan pengguna untuk mengubah RUP sesuai kebutuhan (Kruchten, 1999). Empat fase utama penggunaan RUP yaitu:

- *Inception*

Tahap pertama dari RUP adalah memodelkan proses bisnis dari sistem, kemudian membuat kebutuhan bagi pengembangan. Tahapan *inception* terbagi menjadi dua, yaitu : *business modelling* dan *Requirement*. *Business modeling* sebenarnya tersebar mulai fase *inception*, *elaboration*, dan *construction*. Namun *Business modeling* banyak dilakukan pada fase *inception*. Di dalam *business modelling*, dilakukan pembuatan proses bisnis yang disebut dengan *use case* bisnis. *Use case* bisnis menjamin pemahaman bersama di antara *stakeholder* tentang proses bisnis apa yang perlu didukung di dalam organisasi (*Rational Software*, 1998). *Use case* bisnis dianalisis untuk memahami bagaimana bisnis harus mendukung proses bisnis yang akan terjadi selama siklus operasi perusahaan (*Rational Software*, 1998).

Requirements tersebar dari fase *inception*, *elaboration*, dan *construction*. Namun *requirements* banyak dilakukan di fase *inception*. Tujuan dari *requirements* adalah untuk mendeskripsikan apa yang bisa dilakukan oleh sistem dan memperkenalkan *developer* dan *customer* untuk setuju dengan deskripsi tersebut (*Rational Software*, 1998). Di dalam *requirements*, dibuat daftar kebutuhan fungsional sistem, yang merepresentasikan fitur-fitur utama dari sistem yang akan dibuat. Selain itu, juga dibuat daftar kebutuhan non fungsional, yang merepresentasikan kebutuhan atau spesifikasi tambahan dari sistem yang akan dibuat.

- *Elaboration*

Tahap kedua dari RUP adalah membuat model arsitektur dari sistem, model arsitektur dibuat dari analisis dan desain sistem yang akan dikembangkan. Tahap *elaboration* terbagi menjadi dua, yaitu *analysis* dan *design*. *Analysis & design* tersebar dari fase *inception*, *elaboration*, *construction*, dan *transition*.

Namun *analysis & design* banyak terfokus di fase *elaboration*. Tujuan dari *analysis & design* adalah untuk menunjukkan bagaimana sistem akan direalisasikan nanti (*Rational Software*, 1998). Hasil dari *analysis and design* adalah model desain (opsi sebuah model analisis). Model desain berfungsi sebagai abstraksi dari *source code*, model desain seperti sebuah *blueprint* dari bagaimana *source code* disusun dan ditulis (*Rational Software*, 1998).

- *Construction*

Pada fase ini, pembuatan sistem dimulai, pembuatan dimulai dari komponen dan fitur-fitur dari sistem hingga sistem selesai dikembangkan. Setelah pengembangan selesai, maka dilakukan implementasi dan pengujian sistem untuk melihat performa sistem dan memperbaiki sistem hingga sesuai dengan tujuan awal pengembangan.

- *Transition*

Setelah implementasi dan pengujian selesai, maka tahap terakhir dari RUP adalah instalasi. Sistem yang sudah jadi akan dipasangkan sesuai dengan permintaan klien. Tetapi fase ini tidak hanya sebatas instalasi saja, namun hingga para klien benar-benar dapat menggunakan sistem tersebut, biasanya hal yang dilakukan adalah dengan melakukan pelatihan pada klien atau pemberian tutorial.