

# Kajian Penentuan Alternatif Tester Terbaik Berdasarkan Kompetensi Yang Ditentukan

Joe Lian min<sup>1</sup>, Ani Rahmani<sup>2</sup>, Bambang Wisnuadhi<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung 40012  
E-mail : {joelianmin, anirahma, bwisnu} @jtk.polban.ac.id

## ABSTRAK

Industri perangkat lunak biasanya telah menentukan kompetensi yang harus dimiliki oleh individu yang ingin bergabung diperusahaannya, termasuk kompetensi yang harus dimiliki oleh seorang calon *software tester*. Adanya kebutuhan dalam rekrutmen terhadap *software tester* tersebut, dimungkinkan dilakukan kajian yang berfokus pada bagaimana mendapatkan alternatif *tester* terbaik dari kompetensi yang telah ditentukan. *Simple additive weighting* adalah salah satu metode *multiple-attribute decision making* yang dipilih untuk diterapkan pada kasus menentukan alternatif tester terbaik. Konsep dasar metode ini adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif semua atribut (disebut juga kriteria). Perhitungan bobot yang digunakan pada penelitian ini menggunakan persamaan *rank order centroid*. Ada lima kompetensi yang ditetapkan pada penelitian ini, yakni dilihat dari jumlah *bug* yang ditemukan, kesalahan identifikasi, lama pengerjaan testing, pengalaman kerja, *training / certification* dalam bidang *software testing*, dan latar belakang pendidikan. Data uji coba didapatkan dari eksperimen yang dilakukan terhadap limabelas orang subjek penelitian dari berbagai industri software dan pihak lain yang berprofesi sebagai tester. Subjek penelitian diminta melakukan *end-to-end testing* pada *web online shop* sederhana yang dikembangkan dalam penelitian. Dari hasil eksperimen dapat diketahui bahwa metode *simple additive weighting* (SAW) dapat digunakan untuk mendapatkan alternatif tester terbaik berdasarkan kompetensi yang ditentukan.

### Kata Kunci :

*simple additive weighting, rank order centroid, end-to-end testing, tester competency*

## 1. PENDAHULUAN

Pengujian perangkat lunak (*software testing*) merupakan kegiatan yang harus dilakukan oleh perusahaan pengembang *software* untuk meyakinkan kualitas dari sebuah aplikasi sebelum di-*launching*. *Software testing* adalah sebuah proses mengeksekusi program dengan maksud menemukan error [1]. Terkait hal ini, maka peran seorang *software tester* sangat berpengaruh pada proses *software testing*.

Di perusahaan atau industri *software*, rekrutmen *software tester* biasanya terlebih dahulu ditetapkan kompetensi apa yang harus terpenuhi oleh seorang *software tester* yang ingin bergabung dalam perusahaannya. Pemilihan *tester* akan sangat mudah jika hanya tersedia dua calon, akan tetapi berbeda jika ada banyak *tester* yang melamar. Seiring perkembangan zaman dan untuk mencukupi kebutuhan, terkadang kompetensi yang ditetapkan perusahaan pun terus berkembang.

Bertambahnya kriteria ini pun membuat semakin sulit dalam pengambilan keputusan untuk mendapatkan alternatif *tester* terbaik.

*Multiple attribute decision making* (MADM) adalah cabang pengambilan keputusan yang paling terkenal, dimana salah satu metodenya adalah *Simple Additive Weighted* (SAW). Konsep dasar SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua attribute (disebut juga kriteria). Terdapat beberapa penelitian yang memanfaatkan metode SAW, diantaranya penelitian pada kasus pemilihan makanan alternatif [2]. Penelitian tersebut membandingkan dua metode MADM yaitu SAW dan *weighted product* (WP). Dari hasil penelitian tersebut, metode WP lebih cepat dibanding SAW. Penelitian lainnya yaitu analisis sensitif pada metode SAW karena kebutuhan bahwa kriteria mungkin perlu diubah

oleh pembuat keputusan bergantung pada keadaan saat itu [3].

Beberapa penelitian di atas sangat membantu sebagai penguatan teori penerapan SAW untuk dicobakan dalam menentukan alternatif *tester* terbaik. Sayangnya pada penelitian Adriyendi [2], tidak dipaparkan lebih lanjut mengenai bagaimana menentukan pengukuran cepat yang dimaksud pada penentuan kriteria, selain itu pada penelitian tersebut hanya memaparkan bobot untuk kriteria utama sedangkan bobot kriteria terukur tidak dipaparkan sehingga pembangunan bobot untuk *decision matrix* tidak terlihat. Hal ini menyebabkan pada penelitian ini, penentuan alternatif *tester* terbaik masih diuji coba pada penerapan metode SAW tidak langsung menggunakan metode WP, karena metode SAW paling dikenal dan banyak digunakan untuk pengambilan keputusan dengan beberapa atribut.

## 2. MATERIAL DAN METODE

Penelitian diawali dengan melakukan kajian metode yang kemungkinan dapat diterapkan untuk mendapatkan alternatif *tester* terbaik. Setelah itu menetapkan kompetensi yang akan diuji coba. Dalam hal ini diasumsikan *tester* dilihat dari hasil *test report* yang didapat setelah melakukan *end-to-end testing* sebagai kompetensi. Selain dilihat dari hasil *test report* yang dikerjakan, juga dilihat dari faktor-faktor lain pendukung seorang *software tester*. Hal berikutnya adalah melakukan eksperimen untuk mendapatkan data uji coba *riil* yang kemudian diolah untuk diuji coba dalam penerapan metode SAW. Selanjutnya dibuatkan sebuah *prototype* aplikasi, apabila alternatif *tester* terbaik dari berbagai kompetensi dapat ditentukan dengan penerapan metode tersebut.

Beberapa materi utama serta tahapan penting dalam penelitian dideskripsikan singkat dibawah ini.

### 2.1 End-to-end testing

*End-to-end testing* merupakan salah satu teknik testing yang harus dilakukan oleh perusahaan pengembang web, karena setelah melewati *end-to-end testing* secara umum telah menjamin interaksi *user* dengan halaman web sudah sesuai dengan kebutuhan *user* [4]. Pelaksanaan *end-to-end testing* terbagi menjadi dua metode,

yaitu *horizontal end-to-end testing* dan *vertical end-to-end testing*.

*Horizontal end-to-end testing* merupakan metode yang paling banyak digunakan. Jika ingin melakukan *end-to-end testing* dengan metode horizontal sebagai contoh pada *web* penjualan buku, maka semua proses yang mencakup pengisian data pembeli, detail pembelian buku termasuk detail pembelian yang dilakukan pembeli harus dilakukan testing dari awal sampai akhir.

*Vertical end-to-end testing* merupakan metode yang sangat kompleks sehingga jarang dilakukan, karena *end-to-end testing* dengan metode ini melakukan testing termasuk *application programming interface* (API) dan *structured query language* (SQL).

Pelaksanaan *end-to-end testing* dapat menggunakan salah satu metode ataupun keduanya untuk kebutuhan testing skenario kompleks [5]. Pada pelaksanaannya, *end-to-end testing* merupakan salah satu teknik testing yang bisa dieksekusi dengan *manual testing* ataupun dengan *automated software testing*.

### 2.2 Rank Order Centroid (ROC)

Pembobotan diperlukan dalam penerapan metode SAW untuk pembobotan kriteria. Terdapat beberapa pendekatan dalam pembobotan berdasarkan ranking dari atribut dari yang paling penting sampai yang kurang penting. Metode pembobotan tersebut antara lain *rank sum* (RS) *weights*, *rank reciprocal* (RR) *weights*, dan *rank order centroid* (ROC) *weights*.

Metode yang paling baik dari segi *performance* adalah ROC [6]. Didalam [6] juga dijelaskan bahwa pembobotan ROC diperkenalkan oleh Barron dan Barret (1996) dengan rumus:

$$w_i = \frac{1}{n} \sum_{j=1}^n \frac{1}{j}, \quad i = 1, 2, \dots, n \quad (1)$$

ROC disebut juga pembobotan *centroid* (yaitu titik pusat massa), dimana untuk mengidentifikasi satu set bobot yang mewakili semua kombinasi bobot yang memungkinkan dapat diterima dan konsisten dengan batasan ketimpangan linear yang ditetapkan pada bobot.  $K = \{w: w_1 \geq w_2 \geq \dots \geq w_n, \sum_{i=1}^n w_i = 1, w_i \geq 0, i = 1, \dots, n\}$

Titik ekstrim di K direpresentasikan oleh  $ext^i = (\frac{1}{i}, \frac{1}{i}, \dots, \frac{1}{i}, 0, \dots, 0), i = 1, \dots, n$ .

Dimana  $ext^i$  adalah titik ekstrim ke i dengan i element positif dan n-1 nol.

Edwards dan Barron (1994) kemudian menyediakan *straightforward formula* untuk menentukan sebuah titik pusat massa (*centroid point*) di K, dengan koordinat rata-rata titik ekstrim (persamaan 2)

$$w_1^{ROC} = \frac{1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}}{n}, w_2^{ROC} = \frac{0 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}}{n}, \dots, w_n^{ROC} = \frac{0 + 0 + 0 + \dots + 0 + \frac{1}{n}}{n} \quad (2)$$

$w_1^{ROC}$  berhubungan dengan atribut yang paling penting,  $w_2^{ROC}$  berhubungan dengan atribut kedua yang penting, dan begitu seterusnya. Berdasarkan uraian di atas, dapat diketahui bahwa dari penggunaan ROC, jika  $C_1 \geq C_2 \geq \dots \geq C_n$  maka  $W_1 \geq W_2 \geq \dots \geq W_n$  dengan  $\sum_{i=1}^n w_i = 1$ .

### 2.3 Simple Additive Weighting (SAW)

Proses pengambilan keputusan melibatkan serangkaian langkah mengidentifikasi masalah, membangun preferensi, mengevaluasi alternatif dan menentukan alternatif terbaik [7]. Pengambilan keputusan dengan hanya mempertimbangkan kriteria tunggal lebih mudah dilakukan oleh pembuat keputusan, karena dalam memilih alternatif hanya perlu melihat nilai yang lebih tinggi terhadap kriteria tersebut. Tetapi pada saat ini seiring berkembangnya zaman ada banyak kriteria. Bertambahnya kriteria membuat sulit mengambil keputusan untuk memutuskan alternatif yang terbaik. Menyikapi hal ini, saat ini banyak dikembangkan metode untuk menghadapi hal tersebut.

*Multiple criteria decision making* (MCDM) adalah sebuah disiplin yang muncul yang mendukung pengambilan keputusan menghadapi beberapa kriteria yang biasanya bertentangan [8]. Masalah MCDM dapat dikelompokkan menjadi dua kategori utama yaitu *multiple attribute decision making* (MADM) dan *multiple objective decision making* (MODM). Pengelompokan tersebut untuk memfasilitasi penelitian sistematis dalam bidang MCDM.

SAW merupakan salah satu metode MADM yang paling dikenal dan banyak digunakan untuk pengambilan keputusan dengan beberapa atribut, dikenal juga dengan *weighted sum model* (WSM) atau *scoring method* (SM).

Formulasi perhitungan dengan metode SAW adalah sebagai berikut [3].

$$S_i = \sum_{j=1}^M w_j r_{ij} \quad (3)$$

Untuk  $i = 1, 2, \dots, N$ , dimana :

$S_i$  merupakan keseluruhan skor alternatif ke-i. Sedangkan  $r_{ij}$  adalah rating atau peringkat normal alternatif ke i untuk kriteria j, dan  $w_j$  adalah pentingnya (bobot) kriteria ke j. N adalah jumlah alternatif, dan M adalah jumlah kriteria. Sedangkan formulasi perhitungan untuk mendapatkan nilai  $r_{ij}$  adalah sebagai berikut.

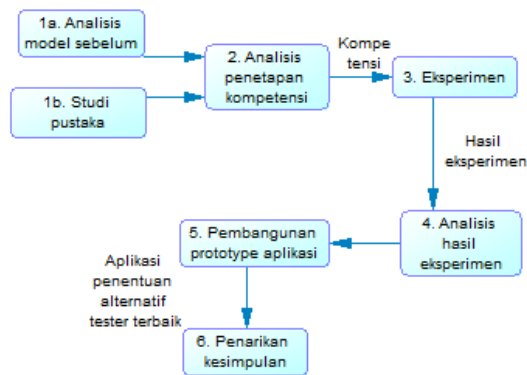
$$r_{ij} = \frac{x_{ij}}{\max_i x_{ij}} \quad (4)$$

$$r_{ij} = \frac{\frac{1}{x_{ij}}}{\max_i (\frac{1}{x_{ij}})} = \frac{\min_i X_{ij}}{X_{ij}} \quad (5)$$

Persamaan 4 digunakan untuk mencari nilai  $r_{ij}$  jika j merupakan atribut *benefit* (untung), dengan  $\max_i X_{ij}$  adalah nilai terbesar dari setiap kriteria i. Sedangkan persamaan 5 digunakan jika j merupakan atribut *cost* (biaya) dengan  $\min_i X_{ij}$  adalah nilai terkecil dari setiap kriteria.

### 2.4 Tahapan Penelitian

Tahapan penelitian yang dilakukan ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

### 3. IMPLEMENTASI

Tahapan penelitian Gambar 1, sampai pada tahap eksperimen diimplementasi sebagai berikut:

#### 3.1 Penetapan Kompetensi

Dalam *working paper* [9], *The Organisation for Economic Cooperation and Development* atau OECD (2001) memberikan definisi tambahan mengenai *human capital* (modal manusia), meliputi dimensi:

1. *Knowledge*  
*Knowledge* atau pengetahuan sebagai kumpulan fakta, prinsip, teori dan praktik. Definisi tersebut mencakup atribut individu yaitu pendidikan formal, hal – hal yang didapatkan di luar sekolah termasuk di dalamnya kesehatan fisik, emosional dan mental.
2. *Skills*  
*Skills* atau keterampilan, sebagai kemampuan untuk menerapkan pengetahuan dan menggunakan pengetahuan untuk menyelesaikan masalah.
3. *Competency*  
*Competency* atau kompetensi, sebagai kemampuan untuk menggunakan pengetahuan dan keterampilan secara tepat dalam konteks kehidupan nyata dan situasi.
4. *Personal attributes*  
*Personal attributes* atau atribut pribadi, yaitu sifat kepribadian, disposisi perilaku dan karakteristik fisik yang mungkin memiliki nilai pada pasar tenaga kerja.

Dari hasil survei dapat diketahui faktor-faktor yang mempengaruhi hasil testing, yaitu [10] :

1. Pengukuran *performance* dari tester dapat dilihat dari faktor :
  - Kualitas *bug* yang ditemukan (92,3%)
  - *Bug advocacy* (83,7%)
  - Ketelitian perencanaan dan pelaksanaan testing (78.8%)
  - *Severity* dari *bug* yang ditemukan (76%)
  - Jumlah *bug* yang ditemukan (walaupun 34% dari responden tidak setuju dengan faktor ini terlalu berpengaruh berpengaruh)

Faktor lain yang berpengaruh untuk pengukuran *performance tester* yaitu kemampuan *programming* dan *academic performance* serta faktor kontributor *performance* seperti positif *attitude*, pengetahuan spesifik mengenai teknik testing, dan lain-lain.

2. Pengaruh *automated testing* (tidak dibahas lebih lanjut karena penelitian ini berfokus pada manual testing)
3. Pengalaman (mayoritas dari responden setuju bahwa faktor ini berpengaruh dalam *software tester*)
4. Karakter personal *software tester* (70.2%)
5. Adanya *training / certification*

Mengacu pada kedua penelitian di atas, dan disesuaikan dengan kebutuhan penelitian serta kebutuhan industri (*software testing* dapat menemukan *bug* sebanyak mungkin sebelum aplikasi di-*launching* dan diharapkan *software tester* tidak salah dalam mengidentifikasi *bug* serta harapan dapat menekan waktu *development* dalam *software testing*), maka kompetensi awal yang digunakan untuk *sampling* penerapan metoda SAW agar mendapatkan alternatif tester terbaik ditunjukkan Tabel 1.

Tabel 1. Kriteria Utama dan Kriteria Pendukung

No	Kriteria Utama	Kriteria Terukur
1	Jumlah <i>bug</i> yang ditemukan ( $C_1$ )	a. 0-5 b. 6-10 c. 11-15 d. 16-20
2	Kesalahan identifikasi ( $C_2$ )	a. 0-5 b. 6-10 c. 11-15

		d. 16-20
3.	Lama pengerjaan testing (C <sub>3</sub> )	a. < 30 menit b. 31 menit – 60 menit c. 61 menit – 120 menit d. > 120 menit
4	Pengalaman kerja (C <sub>4</sub> )	a. 0-1 tahun b. 2-3 tahun c. > 3 tahun
5	Training / Certification (C <sub>5</sub> )	a. Pernah / Ada b. Tidak pernah / Tidak ada
6	Latar belakang pendidikan (C <sub>6</sub> )	a. D3 b. D4 / S1 c. S2

Tabel 2 yang menunjukkan ranking dan bobot kriteria utama dan

Tabel 3, Tabel 4, Tabel 5, Tabel 6 yang menunjukkan ranking dan bobot kriteria terukur untuk masing-masing kriteria utama.

$$w_1^{ROC} = \frac{1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6}}{6} = \frac{2.4503}{6} = 0.41$$

Tabel 2. Rank dan Bobot Kriteria Utama

### 3.2 Penentuan Ranking dan Perhitungan Pembobotan

Langkah selanjutnya adalah melakukan penentuan ranking untuk kriteria utama, diikuti penentuan ranking untuk kriteria terukur. Penentuan *rank* ini dilihat dari kebutuhan industry, dimana agar pada *fase software testing* dapat menemukan *bug* sebanyak mungkin sebelum aplikasi di-*launching* dan diharapkan *software tester* tidak salah dalam mengidentifikasi *bug* serta harapan dapat menekan waktu *development* aplikasi dalam fase *software testing*. Karena hal tersebut, maka secara berurutan *rank* untuk C<sub>1</sub>, C<sub>2</sub>, dan C<sub>3</sub> adalah 1, 2, 3.

Pengalaman kerja subjek penelitian yang terlibat tidak *spesifik* harus dalam bidang *software testing* (posisi *software tester* atau *Quality Assurance*), hal ini karena mempertimbangkan bahwa *performance* pelaksanaan testing juga dipengaruhi oleh faktor tester yang memiliki pengalaman programming [10]. Hal tersebut menyebabkan pengalaman kerja (C<sub>4</sub>) di *rank* ke-4 pada penelitian ini.

Di sisi lain, penentuan *rank* untuk kriteria terukur C<sub>1</sub>, semakin banyak *bug* yang ditemukan maka akan menjadi *rank* 1, untuk kriteria terukur C<sub>2</sub> dan C<sub>3</sub>, semakin sedikit kesalahan dalam identifikasi *bug* dan semakin sedikit waktu yang digunakan untuk testing maka akan menjadi *rank* 1.

Setelah didapatkan ranking untuk kriteria utama, hal selanjutnya melakukan perhitungan bobot. Perhitungan bobot untuk kriteria utama maupun kriteria terukur menggunakan persamaan 2. Berikut contoh perhitungan bobot untuk C<sub>1</sub>, diikuti

Kriteria utama	Rank	Bobot
Jumlah <i>bug</i> yang ditemukan (C <sub>1</sub> )	1	0.41
Kesalahan identifikasi (C <sub>2</sub> )	2	0.24
Lama pengerjaan testing (C <sub>3</sub> )	3	0.16
Pengalaman kerja (C <sub>4</sub> )	4	0.1
Training / Certification (C <sub>5</sub> )	5	0.06
Latar belakang pendidikan (C <sub>6</sub> )	6	0.03

Berdasarkan material ROC, diketahui total keseluruhan bobot kriteria (utama, berlaku juga untuk terukur) harus bernilai 1, maka :

$$0.41 + 0.24 + 0.16 + 0.1 + 0.06 + 0.03 = 1$$

Tabel 3. Kriteria Terukur C<sub>1</sub> dan C<sub>2</sub>

C <sub>1</sub>	Rank	Bobot	C <sub>2</sub>	Rank	Bobot
0 - 5	4	0.06	0 - 5	1	0.52
6 - 10	3	0.15	6 - 10	2	0.27
11 - 15	2	0.27	11 - 15	3	0.15
16 - 20	1	0.52	16 - 20	4	0.06

Tabel 4. Kriteria Terukur C<sub>3</sub>

C <sub>3</sub>	Rank	Bobot
<30 menit	1	0.52
31 menit – 60 menit	2	0.27
61 menit – 120 menit	3	0.15
> 120 menit	4	0.06

Tabel 5. Kriteria Terukur C<sub>4</sub> dan C<sub>6</sub>

C <sub>4</sub>	Rank	Bobot	C <sub>6</sub>	Rank	Bobot
0 – 1 tahun	3	0.11	D3	3	0.11
2 – 3 tahun	2	0.28	D4 / S1	2	0.28
> 3 tahun	1	0.61	S2	1	0.61

Tabel 6. Kriteria Terukur C<sub>5</sub>

C <sub>5</sub>	Rank	Bobot
Pernah / Ada	1	0.75
Tidak Pernah / Ada	2	0.25

### 3.3 Eksperimen

Diasumsikan bahwa pada setiap rekrutmen dalam industri selalu diadakan test masuk, untuk itu maka disiapkan eksperimen dimana



subjek penelitian harus melakukan *testing*, dan *test report* hasil *testing* akan diolah. Eksperimen ini dilakukan untuk mendapatkan data uji coba riil untuk mencoba penerapan metode SAW. Jika SAW berhasil diterapkan, maka data uji coba ini selanjutnya digunakan sebagai data untuk menguji prototype aplikasi penentuan *tester* terbaik yang dibuat.

Eksperimen dilakukan dengan melakukan *end-to-end-testing* terhadap objek penelitian yang telah dibuat, yaitu sebuah web online shop sederhana sebagai *system under test* (SUT), dimana SUT tersebut hanya memanfaatkan *fake back-end* untuk pemrosesan data (java script, tanpa adanya database) dan masih memanfaatkan angular js untuk *front-end*. Pada SUT telah ditanam bug. Bug yang dimaksud termasuk *typographical error*, *logic error* dan kesalahan alur skenario (perpindahan URL).

Selain SUT disediakan juga *test cases*, yang memuat alur skenario order kue. Eksperimen dilakukan oleh lima belas subjek penelitian sebagai *tester*, yang berasal dari berbagai industri *software* dan kalangan mahasiswa yang telah terbiasa melakukan *testing*. Sampling tester diambil secara random yang mewakili beberapa kombinasi dari kompetensi yang ditentukan. Karena rentang *tester* yang terlibat bervariasi, maka proses *testing* dilakukan terpisah pada masing-masing *tester* karena kesulitan dalam menentukan waktu pelaksanaan.

Berdasarkan hal tersebut, *test cases* yang dibuat ada dua, yaitu untuk umum dan mahasiswa yang telah terbiasa melakukan *testing*. Perbedaan keduanya hanya di bagian identitas. Identitas umum memuat nama, jumlah bug yang ditemukan, lama pengerjaan, work experience, training / certification, bidang pekerjaan, riwayat pendidikan. Sedangkan untuk mahasiswa memuat nama, jumlah bug ditemukan, lama pengerjaan, project experience (karena mahasiswa biasanya ada yang berkuliah sambil terlibat project sebagai part-timer atau freelance), project part, training / certification, pendidikan yang sedang ditempuh, programming dan *testing experience*.

Setiap tester yang terlibat penelitian diberikan folder yang berisi SUT, *test cases* dan user manual panduan pelaksanaan *end-to-end testing*.

## 4. HASIL DAN PEMBAHASAN

Hasil eksperimen dan pembahasan penerapan metode SAW dijelaskan sebagai berikut.

### 4.1 Hasil Eksperimen

Berdasarkan hasil analisis dan perhitungan yang dilakukan dari lima belas *tester*, hanya empat belas data yang bisa dipakai untuk uji coba penerapan metode SAW. Hal ini karena T11 (tester ke 11), tidak mengisi atau mengukur waktu *start* dan *finish* dalam mengeksekusi satu *test item* dan pengisian *test result* secara keseluruhan (hanya sebagian) sehingga tidak dapat diukur lama pengerjaannya. Pada bagian *sheet* identitas waktu keseluruhan T11 hanya menuliskan 1 hari, hal ini tetap tidak dapat di konversikan apakah 1 hari tersebut 1 hari jam kerja atau 1 hari 24 jam.

Data perolehan eksperimen ditunjukkan pada Tabel 7. Untuk perhitungan kesalahan identifikasi termasuk didalamnya, jika :

- Tester beranggapan bukan *bug* (status : OK), padahal seharusnya *bug* (status : NOK)
- Tester beranggapan hal tersebut *bug* (status : NOK), padahal bukan *bug* (status : OK)

Tabel 7. Data Perolehan Eksperimen

Alternatif	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>
T1	10	10	90	0	Tidak pernah	D4 / S1
T2	12	8	94	0	Tidak pernah	D4 / S1
T3	11	9	112	0	Tidak pernah	D3
T4	12	8	119	6	Tidak pernah	S2
T5	14	6	60	4	Tidak pernah	D3
T6	14	6	35	1.5	Tidak pernah	D4 / S1
T7	11	9	59	2.6	Pernah	D4 / S1
T8	10	10	60	7	Tidak pernah	D4 / S1
T9	14	6	60	2	Tidak pernah	D3
T10	14	6	48	2	Pernah	D3
T11	6	10		3	Tidak pernah	D4 / S1
T12	11	9	69	0	Tidak pernah	D4 / S1
T13	14	6	76	0	Tidak pernah	D4 / S1
T14	11	9	52	0	Tidak pernah	D4 / S1
T15	13	7	84	0	Tidak pernah	D4 / S1

### 4.2 Penerapan Metode SAW

#### 4.2.1 Decision Matrix

Tabel 8 menampilkan decision matrix, yang merupakan rating untuk setiap alternatif (berdasarkan data perolehan eksperimen) pada setiap kriteria terukur yang telah ditentukan.

Tabel 8. Decision Matrix

0.15	0.27	0.15	0.11	0.25	0.28
------	------	------	------	------	------

0.27	0.27	0.15	0.11	0.25	0.28
0.27	0.27	0.15	0.11	0.25	0.11
0.27	0.27	0.15	0.61	0.25	0.61
0.27	0.27	0.27	0.61	0.25	0.11
0.27	0.27	0.27	0.28	0.25	0.28
0.27	0.27	0.27	0.28	0.75	0.28
0.15	0.27	0.27	0.61	0.25	0.28
0.27	0.27	0.27	0.28	0.25	0.11
0.27	0.27	0.27	0.28	0.75	0.11
0.27	0.27	0.15	0.11	0.25	0.28
0.27	0.27	0.15	0.11	0.25	0.28
0.27	0.27	0.27	0.11	0.25	0.28
0.27	0.27	0.15	0.11	0.25	0.28

#### 4.2.2 Normalisasi Decision Matrix

Setelah membentuk *decision matrix*, langkah selanjutnya adalah melakukan normalisasi *decision matrix*. Langkah yang dilakukan untuk menyusun normalisasi *decision matrix* adalah sebagai berikut :

- Menentukan *atribut* atau *criteria* yang termasuk *benefit* (untung) dan *cost* (biaya).  
Berdasarkan keenam kriteria yang digunakan pada penelitian ini, maka yang digunakan sebagai kriteria *benefit* adalah C<sub>1</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub> dan C<sub>6</sub>. Sedangkan C<sub>2</sub> termasuk kriteria *cost*.
- Menentukan nilai maksimum *benefit criteria* dan nilai minimum *cost criteria*.  
Nilai maksimum (max) pada persamaan 4 untuk *benefit criteria* merupakan nilai terbesar dari setiap *criteria*. Sedangkan nilai minimum (min) pada persamaan 5 untuk *cost criteria* merupakan nilai terkecil dari kriteria. Berdasarkan Tabel 8 maka nilai max dan min dapat ditentukan dan dapat dilihat pada Tabel 9.

Tabel 9. Nilai Max dan Min

Kriteria	Max	Min
C <sub>1</sub>	0.27	-
C <sub>2</sub>	-	0.27
C <sub>3</sub>	0.27	-
C <sub>4</sub>	0.61	-
C <sub>5</sub>	0.75	-
C <sub>6</sub>	0.61	-

- Normalisasi rating *decision matrix*  
Contoh perhitungan nilai r untuk *benefit criteria* menggunakan persamaan 4.

$$r_{11} = \frac{x_{11}}{\max x_{11}} = \frac{0.15}{0.27} = 0.556$$

Sedangkan perhitungan nilai r untuk *cost criteria* menggunakan persamaan 5 sebagai berikut.

$$r_{21} = \frac{\min x_{21}}{x_{21}} = \frac{0.27}{0.27} = 1$$

Tabel 10 menampilkan normalisasi rating *decision matrix* yang sudah dihitung untuk setiap alternatif.

Tabel 10. Nilai Rating (r)

0.556	1	0.556	0.18	0.33	0.459
1	1	0.556	0.18	0.33	0.459
1	1	0.556	0.18	0.33	0.18
1	1	0.556	1	0.33	1
1	1	1	1	0.33	0.18
1	1	1	0.459	0.33	0.459
1	1	1	0.459	1	0.459
0.556	1	1	1	0.33	0.459
1	1	1	0.459	0.33	0.18
1	1	1	0.459	1	0.18
1	1	0.556	0.18	0.33	0.459
1	1	0.556	0.18	0.33	0.459
1	1	1	0.18	0.33	0.459
1	1	0.556	0.18	0.33	0.459

- Menentukan alternatif tester terbaik  
Setelah menghitung nilai r masing-masing alternatif, langkah selanjutnya adalah mendapatkan nilai keseluruhan skor alternatif tester dengan cara mengalikan dengan bobot kriteria yang telah ditentukan.  
Tabel 2. mengacu pada persamaan 3, maka untuk mendapatkan nilai keseluruhan untuk tester T1 adalah sebagai berikut.

$$S_1 = (0.41 * 0.556) + (0.24 * 1) + (0.16 * 0.556) + (0.1 * 0.18) + (0.06 * 0.33) + (0.03 * 0.459) = 0.608$$

Tabel 11 menampilkan nilai keseluruhan untuk semua tester beserta peringkat yang didapatkan dari perhitungan. Dari table tersebut juga diketahui bahwa yang memiliki nilai S tertinggi adalah T5 yaitu sebesar 0.935.

Tabel 11. Peringkat Alternatif Tester

Tester	S	Peringkat
T1	0.608	11
T2	0.791	8
T3	0.782	9
T4	0.889	5
T5	0.935	1
T6	0.890	4
T7	0.929	2
T8	0.761	10
T9	0.881	6
T10	0.921	3
T11	0.791	8
T12	0.791	8
T13	0.862	7
T14	0.790	8

Mengacu pada hasil penerapan metode SAW yang ditampilkan pada

Tabel 11, dari hitungan manual tersebut didapat bahwa T5 (tester ke 5) merupakan alternatif tester terbaik. Jika dilihat lebih lanjut pada Tabel 7 diketahui bahwa T5 menemukan paling banyak *bug*, tergolong sedikit salah identifikasi, tidak membutuhkan waktu lama dalam melaksanakan testing. Hasil ini hampir sama dengan alternatif tester kedua yaitu T7. Hal yang membedakan perhitungan nilai keduanya adalah lama pengalaman kerja, jika ditelaah lebih lanjut, T7 pernah mengikuti *training* dan mempunyai riwayat pendidikan lebih tinggi daripada T5. Dari hal ini dapat disimpulkan penentuan *rank criteria* sangat berpengaruh dalam proses penentuan alternatif tester terbaik. Dari fakta tersebut jika metode ini diterapkan di industri, maka pihak industri harus memperhatikan prioritas kriteria yang harus terpenuhi untuk sebuah posisi.

Selain memperhatikan prioritas kriteria, hal lain yang sangat berpengaruh pada penelitian ini adalah rentang yang ditentukan pada kriteria terukur. Hal ini bisa dilihat dari Tabel 8, dimana nilai yang didapatkan dari tester yang

terlibat dalam penelitian tidak jauh berbeda. Ada dua kemungkinan yang teridentifikasi menyebabkan ketidakberagaman nilai yang didapat, yaitu :

1. Faktor tester yang terlibat pada penelitian memang memiliki kemampuan dan kriteria yang hampir sama yang menyebabkan ketidakberagaman.
2. Rentang yang ditentukan pada kriteria terukur terlalu panjang atau pendek.

Dari kedua kemungkinan ini dapat terwakilkan dari contoh dalam  $C_2$ , dimana tester mendapatkan yang sama 0.27 padahal jumlah kesalahan identifikasi masing-masing tester berbeda.

Berdasarkan hitungan manual terhadap data eksperimen, maka metode SAW dapat digunakan untuk menentukan alternatif *tester* terbaik, dipilih dari jumlah tester 15. Pemilihan tester terbaik tersebut didasarkan pada sejumlah kriteria yang ditentukan.

## 5. KESIMPULAN

Berdasarkan percobaan perhitungan SAW yang dilakukan, dan berdasarkan data eksperimen yang didapatkan maka dapat disimpulkan bahwa :

- a. Metode SAW dapat digunakan untuk mendapatkan alternatif tester terbaik dari berbagai kompetensi yang telah ditetapkan.
- b. Dalam penerapan metode SAW harus diperhatikan penentuan kompetensi atau kriteria yang terlibat, termasuk didalamnya penetapan kriteria terukur.

Saran untuk keberlanjutan penelitian ini adalah sebagai berikut.

- a. Agar mendapatkan keragaman data diharapkan tester yang terlibat dapat mencakup seluruh kombinasi tester yang ditetapkan dan proses testing dapat dilakukan secara langsung (tester dikumpulkan, kemudian melakukan testing secara bersamaan) hal ini agar meminimalisir kekurangan dalam pelaksanaan testing.
- b. Membandingkan metode-metode MCDM yang lainnya, seperti SAW dan WPM dan sebagainya. Hal ini bertujuan untuk



mengetahui metode mana yang lebih baik dalam berbagai hal (contoh: *performance*) untuk diterapkan pada aplikasi penentuan alternatif tester terbaik.

## 6. DAFTAR PUSTAKA

- [1] G. J. Myers, T. Badgett and C. Sandler, *The Art Of Software Testing*, 3 ed., Canada: John Wiley & Son, Inc, 2012.
- [2] Adriyendi, "Multi-Attribute Decision Making Using Simple Additive Weighting and Weighted Product in Food Choice," *MECS*, pp. 8-14, 2015.
- [3] A. Memariani, A. Amini and A. Alinezhad, "Sensitivity Analysis of Simple Additive Weighting Method (SAW): The Results of Change in the Weight of One Attribute on the Final Ranking of Alternatives," *Journal of Industrial Engineering 4*, pp. 13-18, 2009.
- [4] T. Palmer and M. Waltre, "Automated End-to-end User Testing On Single Page Web Applications," 2015.
- [5] Exforsys, "What is End-to-end Testing," 2 August 2011. [Online]. Available: <http://www.exforsys.com/tutorials/testing-types/end-to-end-testing.html>. [Accessed 16 October 2017].
- [6] B. S. Ahn, "Compatible Weighting Method With Rank Order Centroid: Maximum Entropy Ordered Weighted Averaging Approach," *European Journal of Operational Research*, pp. 552-559, 2011.
- [7] G.-H. Tzeng and J. H. Huang, *Multiple Attribute Decision Making Methods and Applications*, Boca Raton: CRC Press, 2011.
- [8] A. Karami, "Utilization and Comparison of Multi Attribute Decision Making Techniques to Rank Bayesian Network Options," 2011.
- [9] Eurostat, "Statistical Approaches to The Measurement of Skills," 2016.
- [10] Kanij, Merkel, & Grundy. Preliminary Survey of Factors Affecting Software. 23<sup>rd</sup> Australian Software Engineering Conference. 2014.