

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Keaslian Penelitian

Telah ada beberapa penelitian yang dilakukan terkait dengan Analisa penggabungan *Metode Profile Matching* dan *Composite Performance Index* dalam pemilihan *driver* terbaik, diantaranya adalah :

No	Thn	Penulis	Judul	Metode	Hasil Penelitian
1	Prosiding SENSEI 2017- Fakultas Teknik Unmuh Jember	Thomas Adi Oktavianus, Wiwik Suharso	Penerapan Metode SAW untuk evaluasi dan penilaian driver berprestasi di perusahaan distribusi	Simple Additive Weighting( SAW)	Pada Penerapan Metode SAW untuk evaluasi dan penilaian driver berprestasi yang telah dilakukan, diketahui program yang dibuat berhasil menentukan driver berprestasi dengan melakukan perangkingan. Namun tingkat kecocokan dengan hasil penilaian driver yang dilakukan oleh perusahaan hanya sebesar 38% yang artinya dari 16 driver yang terpilih versi perhitungan program, hanya 6 driver yang cocok dengan driver berprestasi yang ditentukan perusahaan. Hal ini karena kriteria ketertiban yang

					memiliki nilai bobot cukup besar pada penilaian di program ini tidak begitu diperhatikan pada penilaian yang berjalan.
2	Seminar Nasoinal Inovasi dan Tren (SNIT) 2018 ISBN: 978-602-61268-5-6	Ahmad Sinun, Nurmalasari, Siti Nurajizah, Impin Atmaja	Sistem Penunjang Keputusan Pemilihan Taxi Online Terbaik Menggunakan Metode Analytical Hierarchy Process	Analytical Hierarchy Proses (AHP)	Hasil perhitungan menggunakan AHP untuk menentukan prioritas pilihan sangat bergantung pada pemberian bobot nilai terhadap kriteria dan sub kriteria. Keputusan memberikan nilai dengan prioritas pertama terpilih untuk taxi online adalah Uber-Car dengan nilai skala score prioritas keputusan 0,451 dan disusul berikutnya oleh taxi online Grab-Car dan Go-Car dengan masing-masing score prioritas keputusan 0,333 dan 0,216.
3	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, No. 12, Desember 2017, hlm. 1788-1796	Aditya Sudarmadi, Edy Santoso, Sutrisno	Sistem Pendukung Keputusan Pemilihan Personel Homeband Universitas Brawijaya Menggunakan Metode Profile Matching	Profile Matching (PM)	Metode Profile Matching berhasil diimplementasikan terhadap kasus pemilihan Homeband Brawijaya yang mencari 6 posisi instrument dimana setiap posisi masing-masing memiliki 6 faktor penilaian yaitu 4

					core factor dan 2 secondary factor. Akurasi sistem mencapai 83.3%. Sistem pendukung keputusan pemilihan personel Homeband Universitas Brawijaya memiliki akurasi yang baik dan dapat dijadikan alat
4	Jurnal Komputer Terapan Vol. 2, No. 2, November 2016, 105-116	Andri Anto Tri Susilo, Mona Anggrelika Putri	Sistem Pendukung Keputusan Pemberian Penghargaan kepada Karyawan Bandar Udara Silampari Lubuklinggau Menggunakan Metode Composite Performance Index (CPI)	Metode Composite Performance Index (CPI)	Metode Composite Performance Index dengan Indikator : pendekatan dan pemahaman, perilaku, kerapian, absensi dan tanggung jawab menghasilkan data yang valid. Sistem pendukung keputusan ini maka pimpinan bisa menggabungkan analisa pribadi dan data hasil untuk menghasilkan keputusan yang tepat pada proses pemberian penghargaan karyawan.
5	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, No. 11, November	Oktavianis Kartikasari, Nurul Hidayat, Suprpto	Implementasi Metode Profile Matching Dengan Weight Product Pada Seleksi Staf Operator	Profil Matching dan Weight Product	Hasil perhitungan menggunakan metode Profile Matching dan Weight Product berdasarkan pengujian yang dilakukan yaitu pengujian akurasi adalah 80% di

	2017, hlm. 1331-1336		Komputer (Studi Kasus : UPDT Pendidikan Kecamatan Tambakboyo Kabupaten Tuban Jawa Timur	mana jumlah peringkat yang sama antara hasil sistem dan hasil seleksi staf operator computer adalah 21 data. Berdasarkan pengujian yang dilakukan didapatkan bahwa bobot profil ideal dan profil ideal mempengaruhi proses.
--	----------------------	--	---	---

Perbedaan penelitian yang penulis lakukan dengan penelitian sebelumnya adalah penulis menggunakan penggabungan metode *Profile Matching* dan *Composite Performance Index* dalam pemilihan *driver* terbaik. Hasil dari penelitian ini dapat menjadi solusi yang optimal dan akurat bagi PT. Grab Indonesia dalam pengambilan keputusan pemilihan *driver* terbaik.

## II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (*respositori* pengetahuan *domain* masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan). (Novriansyah, 2015 : 1).

Menurut Novriansyah (2015), karakteristik sistem pendukung keputusan adalah sebagai berikut :

1. Mendukung proses pengambilan keputusan dalam suatu perusahaan.
2. Adanya *interface* manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai kebutuhan.
5. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
6. Memiliki dua komponen utama yaitu data dan model.

### **II.3. Metode *Profile Matching***

Model Pencocokan Profil (*Profile Matching*) adalah suatu proses yang sangat penting dalam manajemen *SDM* dimana terlebih dahulu ditentukan kompetensi (kemampuan) yang dibutuhkan agar diperoleh hasil maksimum dari beberapa kriteria. Kompetensi/kemampuan tersebut haruslah dapat dipenuhi secara maksimal ataupun paling mendekati oleh para *driver*.

Garis besarnya, *profile matching* membandingkan nilai yang didapatkan oleh satu profil dengan standar yang telah ditentukan. Proses membandingkan ini dinamakan gap, dimana semakin kecil nilai gap yang diperoleh maka besar peluang untuk mendapatkan bonus dari perusahaan.

Langkah-langkah dalam melakukan metode ini adalah:

1. Menentukan tujuan dan alternatif yang akan dipilih.
2. Menentukan aspek dan subaspek yang digunakan untuk penilaian.
3. Melakukan pemetaan *Gap*

*Gap* yang dimaksud adalah perbedaan antara profil jabatan dengan profil karyawan atau bisa ditunjukkan pada rumus di bawah ini:

$Gap = \text{Profil pribadi} - \text{Profil kepribadian}$

4. Melakukan perhitungan *core factor* dan *secondary factor*

Setelah menentukan bobot nilai *Gap* untuk ketiga aspek, yaitu aspek kapasitas intelektual, sikap kerja dan perilaku dengan cara sama, setiap kriteria atau aspek dikelompokkan menjadi dua kelompok yaitu kelompok *core factor* dan *secondary factor*.

- a. Perhitungan *core factor* menggunakan rumus dibawah ini:

$$NRC = \frac{\sum NC}{\sum IC} \dots\dots\dots$$

dimana:

NRC = Nilai rata-rata *core factor* tiap aspek

NC = Jumlah total nilai *core factor* tiap aspek

IC = Jumlah item tiap aspek

- b. Untuk menghitung nilai *Secondary Factor* digunakan rumus:

$$NRS = \frac{\sum NS}{\sum IS} \dots\dots\dots$$

dimana:

NRS = Nilai rata-rata *secondary factor* tiap aspek

NS = Jumlah total nilai *secondary factor* tiap aspek

IS = Jumlah item tiap aspek

c. Untuk menghitung nilai total digunakan rumus :

$$60\% (CF) + 40\% (SF) \dots\dots\dots$$

#### 5. Perhitungan Nilai Total

Dari hasil setiap aspek di atas berikutnya dihitung nilai total berdasarkan presentasi dari nilai *core factor* dan *secondary factor* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil.

#### 6. Perhitungan Nilai Rangking

Hasil akhir dari proses *profile matching* adalah ranking dari kepribadian yang dominan dari setiap alternatif. Penentuan ranking mengacu pada hasil perhitungan tertentu. Perhitungan tersebut dapat ditunjukkan pada rumus :

$$Ranking = A\%(K1) + B\%(K2) + C\%(K3) + D\%(K4) + E\%(K5) + F\%(K6) + G\%(K7) + H\%(K8) \dots\dots\dots (6)$$

Dimana :

K = Nilai kriteria dan menyimpan data, maka data harus dalam satu basis data; dibuat satu format sehingga mudah membuat program aplikasinya.

#### II.4. Metode *Composite Performance Index*

Metode *Composite Performance Index (CPI)* merupakan indeks gabungan *Composite Performance Index* yang dapat digunakan untuk menentukan penilaian atau peringkat yang digunakan untuk menentukan penilaian atau peringkat dari berbagai alternative berdasarkan beberapa kriteria (Susilo, 2017 : 3). Prosedur penyelesaian metode *CPI* (Susilo, 2017 : 3) antara lain :

1. Identifikasi kriteria tren positif (semakin tinggi nilainya semakin baik) dan tren negatif (semakin rendah nilainya semakin baik).
2. Untuk kriteria tren positif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi lebih tinggi.
3. Untuk kriteria tren negatif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi secara proporsional lebih rendah.

Perhitungan *CPI* dapat dilihat seperti berikut:

$$A_{ij} = (X_{ij}(\min) / X_{ij}(\min)) \times 100$$

$$A_{(i+1,j)} = (X_{(i+1,j)} / X_{ij}(\min)) \times 100$$

$$I_{ij} = A_{ij} \times P_j$$

$$I_i = \sum_{j=1}^n I_{ij}$$

Keterangan :

$A_{ij}$  : Nilai alternatif ke- $i$  pada kriteria ke- $j$

$X_{ij}(\min)$  : Nilai alternatif ke- $i$  pada kriteria awal minimum ke- $j$

$A_{(i+1,j)}$  : Nilai alternatif ke- $i+1$  pada kriteria ke- $j$

$X_{(i+1,j)}$  : Nilai alternatif ke- $i+1$  pada kriteria awal ke- $j$

$P_j$  : Bobot kepentingan kriteria ke- $j$

$I_{ij}$  : Index alternatif ke- $i$

$I_i$  : Indeks gabungan kriteria alternatif ke- $i$

$I$  : 1,2,3,..... $n$  ;  $J$ : 1,2,3,..... $m$

## II.5. Unified Modelling Language (UML)


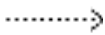




*Unified Modeling Language (UML)* adalah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang di bangun dengan menggunakan teknik pemrograman berorientasi objek (Aprianti, 2016 : 2).


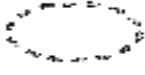
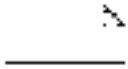
Diagram-diagram yang terdapat dalam *UML* sangat banyak, berikut ini beberapa diagram yang sering di gunakan dalam pengembangan sistem yaitu :

### 1. Use Case Diagram

*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan di buat (Aprianti, 2016 : 3). Simbol-simbol *Use Case Diagram* dapat di lihat pada Tabel II.1 berikut :

**Tabel II.1. Simbol Use Case Diagram**

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya.
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.




	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi di jalankan dan mencerminkan suatu sumber daya


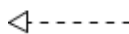
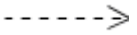

(Sumber : Aprianti ; 2016)

## 2. Class Diagram

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Aprianti, 2016 : 2). *Class Diagram* dapat di lihat pada Tabel II.2 berikut :

**Tabel II.2. Simbol *Class Diagram***

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.



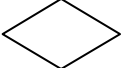
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang di tampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar di lakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya



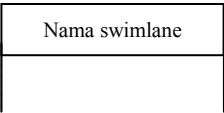
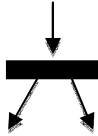
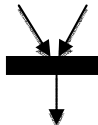
(Sumber : Aprianti ; 2016)

### 3. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Aprianti, 2016 : 3). *Acitivity diagram* dapat di lihat pada Tabel II.3 berikut :

**Tabel II.3. Simbol Activity Diagram**

Simbol	Fungsi	Keterangan
	Status awal	Status awal aktivitas system, sebuah diagram aktivitas memiliki status awal
	Aktivitas	Aktivitas yang di lakukan system, aktivitas biasanya diawali dengan kata kerja
	Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

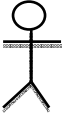
	Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas di gabungkan menjadi satu
	Status akhir	Status akhir yang di lakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	<i>Fork</i>	Di gunakan untuk menunjukan kegiatan yang dilakukan secara parallel
	<i>Join</i>	Di gunakan untuk menunjukan kegiatan yang digabungkan

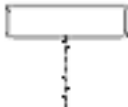


(Sumber : Aprianti ; 2016)

#### 4. *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang di kirimkan dan di terima antar objek. *Sequence Diagram* dapat di lihat pada Tabel II.4 berikut :

**Tabel II.4. Simbol *Sequence Diagram***

Simbol	Fungsi	Keterangan
	<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem di tempatkan di bagian

		atas diagram.
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

(Sumber : Aprianti ; 2016)

## II.6. Normalisasi

Normalisasi adalah salah satu cara untuk meminimalisir pengulangan data (*data redundancy*), normalisasi akan diperlukan jika ada indikasi bahwa tabel yang kita buat tidak baik (terjadi pengulangan informasi, potensi inkonsistensi data pada operasi perubahan, tersembunyinya informasi tertentu dan lain sebagainya) dan diperlukan supaya jika tabel-tabel yang didekomposisi kita gabung kembali dapat menghasilkan tabel awal sebelum didekomposisi, sehingga diperoleh tabel yang baik. Hasil dari normalisasi adalah himpunan-himpunan data (*tabel-tabel*) dalam bentuk normal (*normal form*) (Mulyani, 2016:132).

## **II.7. Basis Data (*Database*)**

*Database* adalah kumpulan dari semua data yang di perlukan oleh sistem. Dengan menggunakan *database*, beberapa aplikasi berbeda bisa saling terintegrasi (Mulyani, 2016 : 148).

*Database Management System* adalah sebuah *software computer* yang digunakan untuk mengolah, mengupdate serta melakukan manipulasi terhadap database dan bertindak sebagai interface (penghubung) antara database dengan aplikasi (Mulyani, 2016:170).

## **II.8. *Microsoft Visual Studio 2010***

*Visual Basic 2010* (VB 2010) merupakan salah satu bagian dari produk pemrograman yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. Sebagai produk pengembangan atau *Integrated Development Environment* (IDE) andalan yang dikeluarkan oleh *Microsoft*, *Visual Studio 2010* berisi beberapa IDE pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#*, dan *Visual F#*. Semua IDE tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5* (Mustakim, 2013 : 2).

## **II.9. *Microsoft SQL Server 2008***

*Microsoft SQL Server 2008* adalah aplikasi yang memiliki fungsi untuk menyimpan dan mendapatkan kembali data yang dibutuhkan oleh aplikasi lain.

