



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Human Resource

Sumber Daya Manusia (SDM) merupakan suatu hal yang sangat penting dan harus dimiliki dalam upaya mencapai tujuan organisasi atau perusahaan. Sumber daya manusia merupakan elemen utama organisasi dibandingkan dengan elemen sumber daya yang lain seperti modal, teknologi, karena manusia itu sendiri yang mengendalikan faktor yang lain. Sumber Daya Manusia adalah penduduk yang siap, mau dan mampu memberikan sumbangan terhadap usaha untuk mencapai tujuan organisasi. Dalam ilmu kependudukan, konsep sumber daya manusia ini dapat disejajarkan dengan konsep tenaga kerja yang meliputi angkatan kerja dan bukan angkatan kerja. Angkatan yang bekerja disebut juga dengan pekerja. (Samsuni, 2017)

2.2 Rekrutmen

Rekrutmen merupakan suatu keputusan perencanaan manajemen sumber daya manusia mengenai jumlah karyawan yang dibutuhkan, kapan diperlukan, serta kriteria apa saja yang diperlukan dalam suatu organisasi. Rekrutmen pada dasarnya merupakan usaha untuk mengisi jabatan atau pekerjaan yang kosong di lingkungan suatu organisasi atau perusahaan, untuk itu terdapat dua sumber tenaga kerja yakni sumber dari luar (eksternal) organisasi atau dari dalam (*internal*) organisasi. Penarikan (rekrutmen) pegawai merupakan suatu proses atau tindakan yang dilakukan oleh organisasi untuk mendapatkan tambahan pegawai melalui beberapa

tahapan yang mencakup identifikasi dan evaluasi sumber-sumber penarikan tenaga kerja, menentukan kebutuhan tenaga kerja, proses seleksi, penempatan, dan orientasi tenaga kerja. (Potale, Lengkong, & Moniharapon, 2016)

2.3 Seleksi

Seleksi merupakan bagian materi dari operasional manajemen sumber daya manusia yaitu pengadaan (*procurement*), sedangkan pengadaan itu sendiri terdiri dari: perencanaan, perekrutan, seleksi, penempatan, dan produksi. Proses seleksi merupakan tahap-tahap khusus yang digunakan untuk memutuskan pelamar mana yang akan diterima. Proses tersebut dimulai ketika pelamar kerja dan diakhiri dengan keputusan penerimaan. (Potale, Lengkong, & Moniharapon, 2016)

2.4 Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer yang membantu para pengambil keputusan mengatasi berbagai masalah melalui interaksi langsung dengan sejumlah database dan perangkat lunak analitik. (Khoiriyah, 2019)

Sistem Pendukung Keputusan (SPK) adalah sistem informasi interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semistruktur dan situasi yang tidak terstruktur, dimana tidak seorangpun tahu secara pasti bagaimana keputusan harusnya dibuat. (Khoiriyah, 2019)

Menurut (Khoiriyah, 2019) untuk melakukan pengambilan keputusan tidak bisa langsung ditentukan namun harus dilakukan melalui beberapa tahap pengambilan yaitu:

1. Identifikasi Masalah
2. Pemilihan Metode
3. Pengumpulan data yang dibutuhkan untuk melaksanakan *model* keputusan
4. Mengimplementasikan model tersebut
5. Mengevaluasi sisi positif dan sisi alternatif yang ada
6. Melaksanakan solusi terpilih

2.5 SAW

Metode *Simple Additive Weighting* merupakan metode yang banyak digunakan dalam pengambilan keputusan yang memiliki banyak atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua *rating* alternatif yang ada. (Sonata, 2016)

Metode SAW sering dikenal dengan metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut. Metode SAW membutuhkan proses normalisasi matrik keputusan (X) kesuatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

Metode SAW mengenal adanya 2 (dua) atribut yaitu kriteria keuntungan (*benefit*) dan kriteria biaya (*cost*). *Benefit* adalah ketika kriteria jika lebih besar maka lebih baik. *Cost* adalah ketika kriteria lebih kecil maka hasil akan lebih baik. (Sonata, 2016)

SAW membutuhkan rating kinerja sebagai berikut:

$$r_{ij} \begin{cases} \frac{x_{ij}}{\max x_{ij}} \\ \frac{\min x_{ij}}{x_{ij}} \end{cases}$$

Rumus 2. 1 Normalisasi *Rating* Kinerja

Sumber: (Sonata, 2016)

Keterangan:

r_{ij} = nilai *rating* kinerja ternormalisasi

x_{ij} = nilai atribut yang dimiliki dari setiap kriteria

$\max x_{ij}$ = nilai terbesar dari setiap kriteria i

$\min x_{ij}$ = nilai terkecil dari setiap kriteria i

benefit = jika nilai terbesar adalah terbaik

cost = jika nilai terkecil adalah terbaik

dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Rumus 2. 2 Rumus nilai preferensi alternatif

Sumber: (Sonata, 2016)

Keterangan:

V_i = *Ranking* untuk setiap alternatif

W_j = Nilai bobot dari setiap kriteria

R_{ij} = Nilai *rating* kinerja yang ternormalisasi

Nilai V_i yang lebih besar mengidentifikasi bahwa alternatif (A_i) lebih terpilih.

Langkah-langkah penyelesaian metode Simple Additive Weighting (SAW) adalah sebagai berikut:

- a. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
- b. Menentukan bobot setiap kriteria.
- c. Melakukan penghitungan Normalisasi kriteria.
- d. Melakukan Perangkinan dengan rumus.

2.6 Agile

Perkembangan dalam *methodology* pengembangan perangkat lunak saat ini telah mengalami banyak perubahan untuk menutupi kelemahan dari *methodology-methodology* sebelumnya. Jika kita melihat ke belakang, *methodology* yang digunakan tidak mampu menangani kemungkinan perubahan atau penambahan requirement pada saat proses pengembangan perangkat lunak berlangsung. Hal inilah yang menjadi pendorong munculnya *methodology* baru dalam pengembangan perangkat lunak. Untuk menangani kelemahan tersebut diperkenalkan *methodology* baru pada dekade 90-an, yakni *agile methods*. (Setyoningrum, 2016)

Arti kata *agile* sendiri berarti tangkas, cepat, atau ringan. Agility merupakan *methodology* yang ringan dan cepat dalam pengembangan perangkat lunak. *Agile Alliance* mendefinisikan 12 prinsip untuk mencapai proses yang termasuk dalam *agility*:

1. Prioritas tertinggi adalah memuaskan pelanggan melalui penyerahan awal dan berkelanjutan perangkat lunak yang bernilai.
2. Menerima perubahan *requirements* meskipun perubahan tersebut diminta pada akhir pengembangan.
3. Memberikan perangkat lunak yang sedang dikerjakan dengan sering, beberapa minggu atau beberapa bulan, dengan pilihan waktu yang paling singkat.
4. Pihak bisnis dan pengembang harus bekerja sama setiap hari selama pengembangan berjalan.
5. Bangun proyek dengan individu-individu yang bermotivasi tinggi dengan memberikan lingkungan dan dukungan yang diperlukan, dan mempercayai mereka sepenuhnya untuk menyelesaikan pekerjaannya.
6. *Methodology* yang paling efektif dan efisien dalam menyampaikan informasi kepada tim pengembangan adalah dengan komunikasi langsung *face-to-face*.
7. Perangkat lunak yang dikerjakan merupakan pengukur utama kemajuan.
8. Proses *agile* memberikan proses pengembangan yang bisa ditopang. Sponsor, pengembang, dan *user* harus bisa menjaga ke-konstanan langkah yang tidak pasti.
9. Perhatian yang terus menerus terhadap rancangan dan teknik yang baik meningkatkan *agility*.
10. Kesederhanaan –seni untuk meminimalkan jumlah pekerjaan– adalah penting.

11. Arsitektur, *requirements*, dan rancangan terbaik muncul dari tim yang mengatur sendiri.

12. Pada *interval regular* tertentu, tim merefleksikan bagaimana menjadi lebih efektif, kemudian menyesuaikannya.

Kelebihan *Agile*:

- Aplikasi akan sangat cepat dalam lingkungan produksi.
- Berkurangnya jumlah bug.
- Integrasi kode mulus.
- Feedback terus menerus dari owner.
- Cepat adaptasi untuk perubahan.

Kekurangan *Agile*:

- Kurangnya dokumentasi.
- Kurangnya komitmen untuk mendefinisikan proses produksi dengan baik.

2.7 PHP

PHP atau kependekan dari *Hypertext PreProcessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *Website* dan dapat di tanamkan pada sebuah skrip HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari.

PHP diciptakan untuk mempermudah pengembang *website* dalam menulis halaman *website* dinamis dengan cepat, bahkan lebih dari itu kita dapat mengeksplorasi hal-hal yang luar biasa dengan PHP. Sehingga dengan demikian

PHP sangat cocok untuk para pemula, menengah maupun expert sekalipun (Djaelangkara, Sengkey, & Lantang, 2015).

2.8 Laravel

Laravel merupakan *web application framework* berbasis PHP yang *open source*, menggunakan konsep MVC: *Model View Controller*. Laravel berada dibawah lisensi MIT License, dengan menggunakan GitHub sebagai tempat berbagi kode. (Rahayuda, 2017)

Sebagai sebuah *framework* PHP, laravel hadir sebagai *platform web development* yang bersifat *open source*. Laravel dirancang khusus untuk memudahkan dan mempercepat proses *web development*. (Mustamiin, Ismantohadi, Ghozali, Darsih, & Inara, 2019)

Kelebihan dari Laravel menurut (Karnyoto & Rangan, 2015) adalah sebagai berikut:

- *Database* dapat di kendalikan.
- Memiliki komunitas yang besar sehingga sudah banyak *troubleshooting*.
- Logika pemrograman dirancang untuk memberikan kemudahan dalam pelacakan sebuah pengembangan sistem sehingga dapat membantu menciptakan segala jenis aplikasi besar maupun kecil.

Namun Laravel juga memiliki kekurangan yaitu:

- Memakan memori yang besar.
- Tidak dapat memproses gambar
- Tidak mendukung MongoDB jika diperlukan.

2.9 MySQL

MySQL adalah salah satu dari sekian banyak sistem *database* yang merupakan terobosan solusi yang tepat dalam aplikasi *database*. MySQL merupakan turunan salah satu konsep utama dalam *database* sejak lama yaitu SQL (*Structured Query Language*). (Danny, 2018)

Penggunaan MySQL sangat cocok untuk perusahaan yang baru saja berdiri karena untuk mengoperasikannya lebih sederhana dan dapat menggunakan *software open source* seperti phpMyAdmin. (Junaidi, 2016)

Keunggulan dari *database* Mysql adalah sebagai berikut (Wongso, 2015):

- a. *Portability*, dapat berjalan stabil pada berbagai sistem operasi.
- b. *Multi user*, dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan,
- c. *Performance tuning*, mempunyai kecepatan yang tinggi dalam menangani query,
- d. *Column typers*, memiliki tipe data yang sangat kompleks,
- e. *Comand and function*, memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam query,
- f. *Security*, memiliki beberapa lapisan sekuritas seperti tingkat *subnet mask*, *hotsname*, *privilege*, *user* dengan sistem perizinan yang mendetail serta *Password* yang terenkripsi
- g. *Scalability dan limits*, mampu menangani basis data dalam jumlah besar, dengan jumlah *field* lebih dari 40 juta, 60ribu tabel dan 5 milyar record. Batas *indeks* mencapai 32 buah pertabel

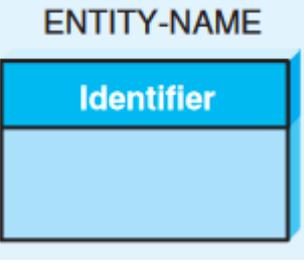
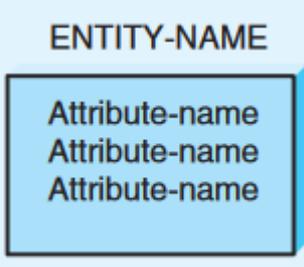
- h. *Localization*, dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari 20 bahasa.
- i. *Connectivity*, dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix Socket*, *Named Pipes*.
- j. *Interface*, memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *API*,
- k. *Client and tools*, dilengkapi dengan berbagai *tools* yang dapat digunakan untuk *Administrasi* basis data sekaligus dokumen petunjuk *online*.
- l. *Structure table*, memiliki struktur tabel yang lebih fleksibel dalam menangani *alter table* dibandingkan dengan *Postgre SQL* dan *Oracle*
- m. Struktur tabel, memiliki struktur tabel yang lebih fleksibel dalam menangani *alter table* dibandingkan dengan *postgre SQL* dan *Oracle*.

2.10 Entity Relationship Diagram

Entity relationship diagram (ERD) adalah sebuah gambar yang menunjukkan informasi yang dibuat, disimpan dan digunakan oleh sebuah sistem bisnis. Seorang analis dapat membaca sebuah ERD untuk menemukan informasi dalam sebuah sistem dan cara mereka saling terhubung. (Dennis, Wixom, & Tegarden, System Analysis Design UML Version 2,0, 2011)

Berikut adalah notasi elemen IDEF1X yang digunakan dalam *usecase diagram*:

Tabel 2. 1 Elemen *Entity Relationship Diagram*

Nama	Penjelasan	Gambar
<i>Entity</i>	Seseorang, tempat atau benda	
<i>Attribute</i>	Bagian dari entity, digunakan oleh minimal satu bisnis proses	
<i>Relationship</i>	Menunjukkan hubungan antara dua entitas	

Sumber: (Dennis, Wixom, & Tegarden, System Analysis Design UML Version 2,0, 2011)

2.11 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah sekumpulan teknik *diagram* untuk memodelkan sebuah sistem. Diagram tersebut terbagi menjadi dua jenis, satu untuk *modeling* struktur sistem dan satunya lagi untuk *modeling* perilaku. Beberapa contoh *modelnya* adalah *usecase diagram*, *activity diagram* dan *class diagram*.

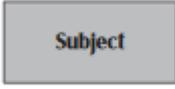
(Dennis, Wixom, & Tegarden, System Analysis and Design An object oriented approach with UML, 2015)

2.11.1 Usecase Diagram

Sebuah *usecase diagram* digunakan untuk lebih baik memahami fungsi sebuah sistem. Dikarekan *usecase diagram* memberikan cara berkomunikasi yang mudah dimengerti agar pengguna dapat sepenuhnya mengerti apa yang sistem akan lakukan, sebuah *usecase diagram* digambarkan saat mengumpulkan dan mendefinisikan keperluan untuk sebuah sistem. (Dennis, Wixom, & Tegarden, System Analysis and Design An object oriented approach with UML, 2015)

Berikut adalah elemen yang digunakan dalam *usecase diagram*:

Tabel 2. 2 Elemen Usecase Diagram

Nama	Penjelasan	Gambar
<i>Actor</i>	Sebuah orang atau sistem yang di beri <i>label</i> untuk sebuah <i>role</i>	 Actor/Role
<i>Use Case</i>	Mewakili fungsi sebuah sistem	 Use Case
<i>Association Relationship</i>	Menghubungkan <i>actor</i> dan <i>use case</i> untuk berinteraksi	
<i>Subject Boundary</i>	Merepresentasikan <i>subject</i> seperti sebuah sistem	 Subject

Sumber: (Dennis, Wixom, & Tegarden, System Analysis Design UML

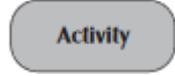
Version 2,0, 2011)

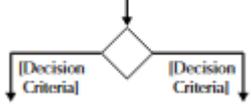
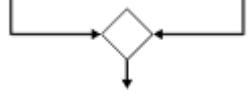
2.11.2 Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku dalam proses bisnis yang tidak bergantung pada objek. *Activity diagram* dapat digunakan untuk memodelkan segala sesuatu mulai dari alur kerja bisnis tingkat tinggi yang melibatkan banyak kasus penggunaan berbeda, hingga *detail* kasus penggunaan individual, sampai ke *detail* spesifik dari metode *individual*. Singkatnya, *activity diagram* dapat digunakan untuk memodelkan segala jenis proses. (Dennis, Wixom, & Tegarden, System Analysis and Design An object oriented approach with UML, 2015)

Berikut adalah elemen yang ada dalam *activity diagram*:

Tabel 2. 3 Tabel elemen Activity Diagram

Nama	Penjelasan	Gambar
<i>Action</i>	Sepotong perilaku	
<i>Activity</i>	Digunakan untuk merepresentasikan sebuah <i>action</i>	
<i>Object Node</i>	Digunakan untuk merepresentasikan objek yang tersambung beberapa <i>object flow</i>	
<i>Control Flow</i>	Menunjukkan urutan eksekusi	
<i>Object Flow</i>	Menunjukkan aliran sebuah object dari satu aktivitas ke aktivitas lainnya	
<i>Initial Node</i>	Menunjukkan awal dari aktivitas	

Nama	Penjelasan	Gambar
<i>Final-Activity Node</i>	Digunakan untuk menghentikan aliran dalam sebuah aktivitas	
<i>Final Flow Node</i>	Digunakan untuk menghentikan sebuah flow spesifik	
<i>Decision Node</i>	Digunakan untuk membuat sebuah <i>test condition</i>	
<i>Merge Node</i>	Digunakan untuk menggabungkan kembali setelah dilakukannya <i>decision node</i>	
<i>Fork Node</i>	Digunakan untuk membagi perilaku menjadi serangkaian kegiatan paralel	
<i>Join Node</i>	Digunakan untuk menyatukan kembali serangkaian kegiatan paralel	
<i>Swimlane</i>	Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom	

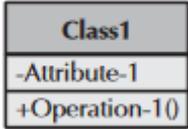
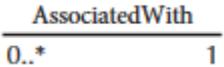
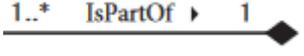
Sumber: (Dennis, Wixom, & Tegarden, System Analysis Design UML Version 2,0, 2011)

2.11.3 Class Diagram

Class diagram adalah *model statis* yang menunjukkan kelas dan hubungan antara kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas, yang mencakup perilaku dan *status*, dengan hubungan antara kelas. (Dennis, Wixom, & Tegarden, System Analysis and Design An object oriented approach with UML, 2015)

Berikut adalah elemen yang ada dalam *class diagram*:

Tabel 2. 4 Tabel elemen *Class Diagram*

Nama	Penjelasan	Gambar
<i>Class</i>	Merepresentasikan seseorang, tempat, atau benda yang perlu ditangkap oleh sistem dan disimpan informasinya	
<i>Attribute</i>	Merepresentasikan properti yang menggambarkan keadaan suatu objek	attribute name /derived attribute name
<i>Operation</i>	Merepresentasikan fungsi dari sebuah <i>class</i>	operation name ()
<i>Association</i>	Merepresentasikan hubungan antar <i>class</i>	
<i>Generalization</i>	Merepresentasikan hubungan antara beberapa <i>class</i>	
<i>Aggregation</i>	Merepresentasikan segi <i>logic</i> dari hubungan antar <i>class</i>	
<i>Composition</i>	Merepresentasikan hubungan fisik dari beberapa <i>class</i>	

Sumber: (Dennis, Wixom, & Tegarden, System Analysis Design

UML Version 2,0, 2011)

2.12. Penelitian Terdahulu

Tabel 2. 5 Tabel Penelitian Sebelumnya

1	Penulis	Kamalia Safitri, Fince Tinus Waruwu, Mesran
	Judul	Sistem Pendukung Keputusan pemilihan karyawan berprestasi dengan menggunakan metode <i>Analytical Hiearachy Process</i>
	Nama jurnal	Media Informatika Budidarma, Vol 1, No 1, Maret 2017
	Kesimpulan	Berdasarkan proses analisis, perancangan dan implementasi pada Penggunaan sistem pendukung keputusan pemilihan karyawan berprestasi di PT. Capella Dinamik Nusantara Takengon dapat diambil kesimpulan bahwa aplikasi dibangun untuk memudahkan dalam pengambilan keputusan karyawan berprestasi
	Kaitan dengan Penelitian	Sebagai bukti bahwa di PT.Capella setelah adanya sebuah sistem, lebih mudah untuk melakukan proses pemilihan karyawan berprestasi di perusahaan tersebut.
2	Penulis	Aji Sasongko, Indah Fitri Astuti, Septya Maharani
	Judul	Pemilihan karyawan dengan metode AHP (<i>Analytic Hierarchy Process</i>)
	Nama jurnal	Jurnal Informatika Mulawarman

	Kesimpulan	Berdasarkan hasil penelitian pemilihan karyawan baru, dapat diambil kesimpulan Proses pembuatan Sistem Pemilihan Karyawan Baru dapat dilakukan dengan metode <i>Analytic Hierarchy Process</i> (AHP) dengan kriteria dan bobot yang telah ditentukan oleh PT. Noreen Surya Perdana yang diperoleh dari hasil wawancara kemudian diproses oleh sistem sehingga menghasilkan output perangkaan karyawan baru.
	Kaitan dengan Penelitian	Panduan untuk bagaimana cara penyusunan laporan
3	Penulis	Ardhi Bagus Primahudi, Fajar Agustinus Suciono, Anag Aris Widodo
	Judul	Sistem pendukung keputusan untuk pemilihan karyawan dengan metode <i>simple additive weighting</i> di PT. Herba Penawar Alwahida Indonesia
	Nama jurnal	Jurnal Informatika Merdeka Pasuruan Vol.2, No.1 Agustus 2016
	Kesimpulan	Sistem pendukung keputusan bertujuan untuk membantu pegawai HRD menentukan pegawai yang sesuai dengan kriteria yang diinginkan dalam bentuk <i>ranking</i> melalui penyeleksian metode SAW.

Kaitan dengan Penelitian	Panduan <i>step by step</i> metode bagaimana cara penggunaan metode SAW
--------------------------	---