

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam tinjauan pustaka di bawah ini terdapat lima referensi dan satu referensi dari penulis

Tabel 2.1 Penelitian Sebelumnya

Komponen	Fauziah Ayu Kusumawardani (2016)	Debora Dwi Putri Ningrum (2012)
Obyek penelitian	Seleksi Calon Karyawan Di PT. Kebon Agung Surabaya	Calon Penerima Beasiswa Peningkatan Pretasi Akademik
Masukan	Data karyawan dan kriteria karyawan	Data Mahasiswa, kriteria syarat beasiswa
Keluaran	Pemilihan Karyawan	Mahasiswa Penerima Beasiswa
Metode	Metode Weighted Product	Metode Weighted Product
Pemrograman	Php	Php

Komponen	Nur Arifah Syafitri (2016)	Penelitian Yashinta Artikasari (2015)	Afthour Riza (2014)	Penelitian yang diajukan (2017)
Obyek penelitian	Pemilihan Laptop Berdasarkan Kebutuhan	Perangkingan Karyawan terbaik	Pemilihan Lokasi Pembangunan Perumahan	Pencocokan kriteria hotel
Masukan	Data laptop dan kriteria spesifikasi kebutuhan	Data karyawan, kriteria kerja	Data lokasi dan Kriteria lokasi	Kriteria hotel dan data hotel
Keluaran	Pemilihan Laptop	Penilaian Karyawan	Pemilihan Lokasi	Permilihan Hotel
Metode	Metode Weighted Product	Metode Weighted Product	Metode Weighted Product	Metode Weighted Product
Pemrograman	Php	Php	Php	Php Codeigniter

Penelitian penggunaan metode weighted product pernah dilakukan oleh mahasiswa Universitas Islam Negeri Maulana Malik Ibrahim Malang yaitu Fauziah Ayu Kusumawardani(2016) dalam skripsi sistem pendukung keputusan seleksi calon karyawan di PT. Kebon Agung Surabaya. Penelitian ini menghasilkan calon karyawan yang akan bekerja di PT. Kebon Agung Surabaya. Kriteria untuk seleksinya yaitu kecakapan, pengalaman, usia, jenis kelamin, pendidikan, syarat-syarat badaniah, penampilan, kecerdasan, kejujuran, bakat, kematangan dan kepribadian. Penelitian yang dilakukan oleh Debora Dwi Putri Ningrum(2012) dalam skripsi sistem pendukung keputusan untuk merangking calon peneriman beasiswa peningkatan prestasi akademik menggunakan metode wp. Penelitian ini

membuat sistem untuk memutuskan penerima beasiswa yang berdasarkan kriteria seperti berikut ipk, pretasi, kegiatan dan penghasilan orang tua.

Selain itu ada sistem penunjang keputusan pemilihan laptop berdasarkan kebutuhan spesifikasi calon pembeli berbasis web, yang dilakukan oleh NurArifiah Syafitri (2016). Penelitian ini menghasilkan daftar laptop yang akan dipilih oleh pembeli dengan spesifikasi kebutuhannya. Kriteria yang digunakan yaitu harga, ram, harddisk, processor dan vga. Yasinta Artikasri (2015) juga pernah melakukan penelitian sistem pendukung keputusan dengan metode wp untuk perangkingan karyawan terbaik. Menghasilkan karyawan terbaik berdasarkan kriteria yang di tetap seperti kinerja pegawai, disiplin, kreatif dan jujur. Afthonur Riza (2014), sistem pendukung keputusan pemilihan lokasi pembangunan perumahan dengan metode weighted product. Pada penelitian ini menghasilkan lokasi yang tempat untuk pembangunan perumahan berdasarkan kriteria seperti jarak pusat kota, jarak jalan utama, jarak pasar dan harga tanah.

Sedangkan penelitan yang saya lakukan untuk mengambil keputusan dalam pemilihan hotel di kota yogyakarta dengan metode weighted product. Sistem ini menghasilkan perangkingan hotel sesuai kriteria yang dipilih oleh pengunjung, kriteria yang digunakan seperti harga, jarak, tipe kamar, tempat tidur dan fasilitas.

2.2 Dasar Teori

Dasar teori di gunakan untuk memahami defines, pengertian dasar dan istilah yang digunakan dalam penelitianini, berikut dasar teori yang di gunakan :

1. Sistem Pendukung Keputusan

Sistem pendukung keputusan merupakan sistem yang berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai masalah yang semi terstruktur dan tidak terstruktur(Kusumadewi,2006).

2. *Multi Attribute Decision Making (MADM)*

Multiple Attribute Decision Making (MADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah MADM, antara lain:

- a. *Simple Additive Weighting (SAW)* yaitu metode penjumlahan terbobot dari rating kinerja pada setiap alternatif atribut.
- b. *Weighted Product (WP)* yaitu metode perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan.
- c. *TOPSIS* yaitu konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negatif.

- d. *Analytic Hierarchy Process* (AHP) yaitu merubah nilai-nilai kualitatif menjadi nilai kuantitatif, sehingga keputusan keputusan yang diambil bisa lebih obyektif.

3. *Metode Weighted Product (WP)*

Weighted Product (WP) merupakan salah satu metode yang digunakan untuk menyelesaikan masalah *Multi Attribute Decision Making* (MADM). Metode *Weighted Product* (WP) menggunakan perkalian untuk menghubungkan nilai atribut (kriteria), dimana nilai setiap atribut (kriteria) harus dipangkatkan dulu dengan bobot atribut (kriteria) yang bersangkutan.

Langkah-langkah dalam menggunakan metode WP sebagai berikut(Kusumadewi,2006).

- a. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
- b. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
- c. Menentukan bobot preferensi(w) tiap kriteria, sebagai berikut.

w_j = bobot kepentingan kriteria j

$\sum w_j$ = jumlah semua bobot kepentingan, dimana w_j berpangkat negatif untuk atribut biaya dan pangkat positif untuk atribut keuntungan.

$$w_j = \frac{w_j}{\sum w_j}$$

- d. Mengalikan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk atribut keuntungan dan bobot berpangkat negatif untuk atribut biaya.

$$S_i = \prod_{j=1}^n X_{ij}^{w_j}$$

Dengan $i=1,2,\dots,m$. Dimana:

S : preferensi alternatif dianalogikan vektor S

X : nilai kriteria

W : bobot kriteria

i : alternatif

j : kriteria

n : banyaknya kriteria

- e. Hasil perkalian tersebut dijumlahkan untuk menghasilkan nilai V untuk setiap alternatif. Preferensi relatif dari setiap alternatif, diberikan sebagai:

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{w_j}}{\prod_{j=1}^n (X_j^*)^{w_j}}$$

Dengan $i=1,2,\dots,m$. Dimana:

S : preferensi alternatif dianalogikan vektor V

X : nilai kriteria

W : bobot kriteria

i : alternatif

j : kriteria

n : banyaknya kriteria

* : banyaknya kriteria yang telah dinilai pada vektor S. (Kusumadewi, 2006)

- f. Mencari nilai alternatif dengan melakukan langkah yang sama seperti pada langkah satu, hanya saja menggunakan nilai tertinggi untuk setiap atribut tertinggi untuk setiap atribut manfaat dan nilai terendah untuk atribut biaya.
- g. Membagi nilai V bagi setiap alternatif dengan nilai standar ($V(A^*)$) yang menghasilkan R.
- h. Mencari nilai alternatif ideal.

4. *Hypertext Preprocessor (PHP)*

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. PHP adalah bahasa *server-side scripting* yang menyatu dengan *Hypertext Markup Language* (HTML) untuk membuat halaman web dinamis. Karena PHP merupakan *server-side scripting* maka sintaks perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membentuk halaman web yang

dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web.

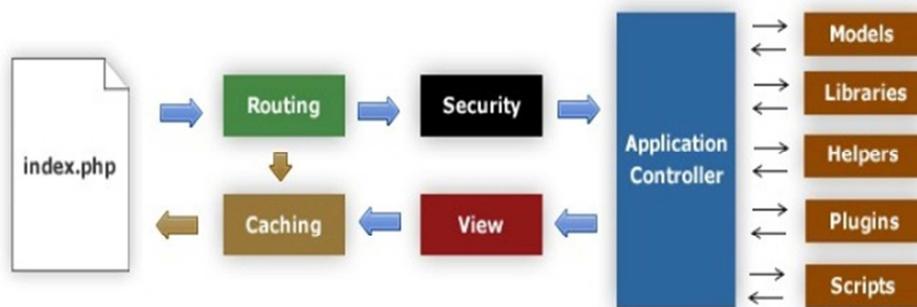
PHP termasuk dalam *Open Source Product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas. PHP dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache dan Xitami. PHP juga mampu lintas *platform*. Artinya PHP dapat berjalan di banyak sistem operasi seperti Windows, Linux, Mac OS serta Solaris. Salah satu keunggulan yang dimiliki oleh PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam *software* sistem manajemen basis data/Database Management System (DBMS) (Kadir, 2008).

5. CodeIgniter Framework

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan pemrograman PHP. *CodeIgniter* memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Tidak seperti beberapa *framework* PHP lainnya, dokumentasi untuk *framework* ini sangat lengkap, yang mencakup seluruh aspek dalam *framework*. *CodeIgniter* juga mampu berjalan pada lingkungan *shared hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat luar biasa (Sidik, 2012).

Dari sisi pemrograman, *CodeIgniter* kompatibel dengan PHP4 dan PHP5, sehingga akan berjalan dengan baik pada *web host* yang banyak dipakai pada saat ini. *CodeIgniter* menggunakan pola desain *Model-View-Controller* (MVC), yang

merupakan cara untuk mengatur aplikasi *web* ke dalam tiga bagian yang berbeda, yaitu *Model* –lapisan abstraksi database, *View* –file-file *template* tampilan depan, dan *Controller* –logika bisnis dari aplikasi. Pada intinya, *CodeIgniter* juga membuat menggunakan ekstensif dari pola desain *Singleton*. Maksudnya adalah cara untuk *me-load class* sehingga jika *class* itu dipanggil dalam beberapa kali, kejadian yang sama pada *class* tersebut akan digunakan kembali. Hal ini sangat berguna dalam koneksi *database*, karena hanya menggunakan satu koneksi setiap kali *class* itu digunakan (Sidik, 2012).



Gambar 2.1 Application Flow Chart CodeIgniter Framework

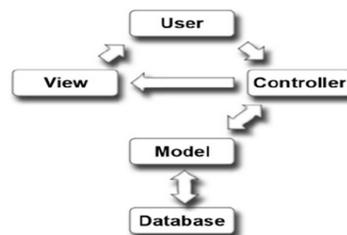
Keterangan:

- Index.php berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan *CodeIgniter*.
- Router memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengannya.
- Jika Cache aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.

- d. Controller memuat *model, core libraries, pligins, helpers*, dan semua *resource* yang diperlukan untuk memproses *request*.
- e. Akhirnya View yang dihasilkan akan dikirim ke *browser*. Jika *cache* aktif, maka *view* akan disimpan sebagai *cache* dahulu, sehingga pada *request* berikutnya langsung ditampilkan.

Model-View-Controller (MVC) pertama kali diperkenalkan peneliti Xerox PARC yang bekerja pada bahasa pemrograman *Smalltalk* diakhir tahun 1970-an dan awal 1980-an. *Smalltalk* adalah bahasa pemrograman yang berorientasi objek, bertipe dinamis, dan efektif, dan hal ini berbeda dari data *mainframe* dan struktur kontrol dalam program *Smalltalk* yang terlibat pada *Windowed User Interfaces*, konsep pemrograman berorientasi objek, pengantar pesan antara komponen-komponen objek, dan kemampuan untuk memonitor dan memodifikasi struktur dan perilakunya sendiri (Sidik, 2012).

Singkatnya, *Model-View-Controller* (MVC), adalah pola desain pengembangan perangkat lunak. MVC adalah sebuah pendekatan untuk memisahkan aplikasi menjadi tiga segmen, yaitu *Models*, *Views*, dan *Controller*. MVC menstrukturisasi aplikasi dengan cara tersebut untuk mempromosikan penggunaan kembali dari kode program (Sidik, 2012).



Gambar 2.2 *ModelView-Controller*

Berdasarkan gambar 2.2, dapat diketahui bahwa ketika datang sebuah permintaan dari *user*, maka permintaan tersebut akan ditangani oleh *Controller*, kemudian *Controller* akan memanggil *Model* jika memang diperlukan operasi *database*. Hasil *query* oleh *Model* kemudian akan dikembalikan ke *Controller*. Selanjutnya *Controller* akan memanggil *View* yang tepat dan mengkombinasikannya dengan hasil *queryModel*. Hasil akhir dari operasi ini akan ditampilkan ke *browser* yang selanjutnya bisa dilihat oleh *user* (Sidik, 2012).

a. *Model*

Model merupakan jenis data yang dapat digunakan adalah *database*, *RSS Feed*, *API calls*, dan setiap tindakan lainnya yang melibatkan pengambilan (*retrieving*), pengembalian (*returning*), memperbarui (*updating*), menghapus (*removing*) data (Sidik, 2012).

b. *View*

File-file yang ditempatkan pada bagian ini bertanggung jawab untuk menunjukkan data kepada para pengunjung situs, atau pengguna dari aplikasi. Tidak ada logika pemrograman, tidak ada *query insert* atau *update* yang harus dijalankan disini, meskipun akses data bisa terjadi dari dua bagian lainnya. Jadi data diambil dari dalam *model* (Sidik, 2012).

c. *Controller*

Controller adalah logika bisnis dari aplikasi. File-file yang ada di CI yang akan melayani sebagai perantara antara *Models* dan *Views*. *Controller* akan

merespon permintaan HTTP dan menghasilkan halaman *web*. *Controller* adalah inti dari aplikasi karena bagian menentukan bagaimana permintaan HTTP harus ditangani (Sidik, 2012).