# BAB II

**LANDASAN TEORI**

## Penelitian Sebelumnya

Pada penelitian yang dilakukan oleh Wulandari (2016) dibagi menjadi 2 manfaat antara lain manfaat teoritis dan manfaat praktis. Manfaat teoritis pada penelitian ini adalah sebagai bahan referensi untuk mahasiswa lain dalam melakukan penelitian yang relevan dan sebagai sarana mengaplikasikan ilmu yang telah didapat di bangku perkuliahan. Manfaat praktis dalam penelitian ini adalah sebagai sarana informasi universitas untuk dapat memperbaiki peringkat *website*nya pada peringkat Webometrics dan sebagai acuan untuk memprediksi peringkat *website* universitas dalam hal ini Universitas Negeri Yogyakarta (UNY) pada rilis Webometrics yang akan datang.

Sehingga dapat disimpulkan mekanisme perhitungan pemeringkatan *website* universitas menggunakan metode pembobotan *Entropy* padapenelitian tersebut terdapat 5 (lima) langkah yaitu (1) normalisasi data, (2) menghitung jumlahnilaidatayangtelahdinormalisasi,(3)perhitungan*Entropy*untuksetiap kriteria ke-i, (4) mencari nilai e(di) untuk masing-masing kriteria, (5) menghitung bobot *Entropy* dan metode pemeringkatan *Promethee* terdapat 7 (tujuh) langkah yaitu (1) normalisasi data, (2) menghitung perbedaan evaluasi alternatif i dengan alternatif lain, (3) menghitung preferensi data, (4) menghitung agregat fungsi preferensi, (5) menentukan nilai *Leaving* dan *Entering Outranking Flow*, (6) menghitung nilai net *Outranking*untuk setiap

10

alternatif, (7) menentukan peringkat semua alternatif. Dan metode pembobotan *Entropy* dan metode pemeringkatan *Promethee* dapat digunakan untuk memprediksi peringkat Webometrics selanjutnya. Hal ini dibuktikan dengan uji statistik Wilcoxon Matched Pairs pada peringkat Webometrics rilis Juli 2016 (Wulandari, 2016).

## Sistem Informasi

T.Cornford (2013) mengatakan sistem informasi menggunakan TIK (Teknologi Informasi dan Komunikasi) diterapkan untuk memperbaiki cara organisasi beroperasi dan untuk membantu meringankan pekerjaan. Hal tersebut dicapai dengan mengumpulkan, menyimpan, serta memproses dan berbagi data dan informasi. Dalam pernyataan tersebut menunjukkan bahwa studi tentang sistem informasi memerlukan empat tujuan yang berbeda tetapi saling berkaitan, diantaranya adalah sebagai berikut:

* + 1. Teknologi digital berbasis komputer,digunakan untuk menangani informasi.
		2. Pengguna menjadi bagian dari sistem informasi.
		3. Menyelesaikan tugas-tugas yang diharapkan untuk kebutuhan dan persyaratan tertentu.
		4. Membangun sistem.

## Metode Simple Additive Weighting (SAW)

Metode Simple Additive Weighting (SAW) sering juga dikenal istilah metode penjumlahan terbobot (Fishburn, 1967). Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (MacCrimmon, 1968).

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut

(Kusumadewi,2006):

𝑥𝑖𝑗

𝑀𝑎𝑥𝑖𝑥𝑖𝑗

𝑟𝑖𝑗=

𝑀𝑖𝑛𝑖𝑥𝑖𝑗

Jika j = atribut keuntungan (benefit)

Jika j = atribut biaya (cost)

…(2.1)

{ 𝑥𝑖𝑗

Dimana :

Rij = Rating kinerjaternormalisasi

Maxij = Nilai maksimum dari setiap baris dan kolom Minij = Nilai minimum dari setiap baris dan kolom Xij = Baris dan kolom dari matriks

Dimana 𝑟𝑖𝑗adalah rating kinerja ternormalisasi dari alternatif Ai pada atributCj;i=1,2,...,mdanj=1,2,...,n.Nilaipreferensiuntuksetiapalternatif(𝑉𝑖) diberikansebagai:

…(2.2)

𝑛

𝑉𝑖=∑𝑤𝑗𝑟𝑖𝑗

𝑗=1

Dimana :

Vi = Nilai akhir dari alternatif

Wi = Bobot yang telah ditentukan Rij = Normalisasi matriks

n = jumlahalternatif

Nilai 𝑉𝑖 yang lebih besar mengindikasikan bahwa alternatif 𝐴𝑖 lebih terpilih. Penilaian nilai standar kriteria dilakukan berpatokan dengan bilangan *fuzzy* seperti dalam Gambar 2.3 berikut ini :



Sumber: Suryadi (2002)

**Gambar 2.3 Grafik Nilai Standar Kriteria**

Keterangan :

SR = Sangat rendah R = Rendah

C = Cukup T =Tinggi

ST = Sangat Tinggi

Langkah penyelesaian SAW menurut Kusumadewi (2006) sebagai

berikut:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiapkriteria.
3. Membuat matriks keputusan berdasarkan kriteria (Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai)sebagai solusi.

## AlexaRank

Alexa Rank merupakan sebuah sistem perangkingan yang dimiliki oleh Alexa Internet, inc. yang mana pengukurannya didasarkan pada tingkat lalu lintas (traffic) dari setiap domain/blog/*website* yang didapatkan dari pengunjung.Jika pengunjung mengunjungi alamat domain/blog/*website* yang sama berkali-kali pada hari yang sama, semua kunjungan akan dihitung sebagai satu. Alexa Rank diperoleh dengan melakukan geometric rata-rata dari jangkauan. Semakin kecil peringkat numeriknya, maka akan semakin baik. Nilai numeriknya bekisar antara 1 hingga 25.000.000 (Purwiantono,2014).

Ada empat *Traffic* yang biasa dipakai untuk menaikkan rangking suatu *website*, antara lain: *Direct Traffic,* adalah trafik yang didapat secara langsung dari pengunjung yang mengetik alamat suatu *website* pada *Address Bar Browser* (Kurnia, 2017). *Website* akan mudah dikenal ketika memakai traffic ini, karena pengunjung yang menuliskan alamat *website* pada mesin pencari, tentu sudah mengenal *website* tersebut.

*Referral Traffic*, berasal dari *website* lain. Hal ini berkaitan dengan link yang terpasang pada *website* lain, baik dilakukan oleh pemilik *website* atau orang lain (Kurnia, 2017). Hasil dari *Traffic* ini didapatkan melalui berapa banyak kunjungan pengguna, yang mengklik link tersebut. *Social Traffic* datang dari sosial media seperti Facebook, Twitter, dan Google Plus (Kurnia, 2017). Hasil dari *Traffic* ini didapatkan dari banyaknya pengguna sosial media yang mengklik link *website*. *Organic Traffic* didapatkan dari mesin pencari, memiliki kualitas paling tinggi dan menjadi tujuan dari *website* di buat (Kurnia, 2017). *Traffic* ini didapatkan dari berapa banyak kunjungan dan *Keyword* pengguna yang menuju halaman *website*, untuk mendapatkan manfaat dari traffic ini, *website* haruslah menuju halaman awal pencarian pada mesin pencari.

## Webometrics

Wulandari (2016) menyebutkan bahwa, Webometrics adalah studi tentang aspek-aspek kuantitatif dari konstruksi dan penggunaan sumber daya informasi, struktur, dan teknologi pada gambar web melalui pendekatan bibliometrik dan informatrik. Dalam melaksanakan penelitian tentang webometrics, studi ini

dilakukan dengan menganalisis dari link dan situs web, evaluasi dari search engine dan studi tentang deskripsi tentangweb. Webometrics yaitu studi tentang konten berbasis web dengan metode kuantitatif dengan tujuan utama untuk penelitian ilmu sosial menggunakan teknik yang tidak khusus untuk satu bidang studi.

Isidro F Aguillo adalah kepala Cybermetrics Lab, sebuah kelompok penelitian Instituto de Politicas y Bienes Públicos (IPP), badan penelitian nasional Spanyol (CSIC). Beliau adalah seorang editor Rangkings Web (Webometrics), termasuk Rangking Web of Universities (sejak2004), Research Centres (sejak 2006), Hospitals, Repositories dan Business Schools (sejak 2008). Beliau adalah penemu dan editor jurnal “Cybermetrics”, jurnal elektronik pertama CSIC. Beliau telah mempublikasikan lebih dari 60 papers dalam jurnal Internasional seperti pada topik sebagai evaluasi kegiatan ilmiah, jurnal elektronik, inisiatif akses terbuka (open access) dan khusus webometrics serta indikator web.

Kriteria-kriteria yang digunakan Webometrics sebagai dasar perhitungan untuk merangking antara lain *Presence*, *Visibility*, *Transparency* dan *Excellence.* Yang mana *Presence* adalah jumlah halaman situs web dan semua jenis file termasuk pdf, *Visibility* yang dimaksudkan di sini adalah jumlah *backlinks* (link yang mengarah ke situs web Perguruan Tinggi), *Transparency* merupakan jumlah sitasi dari *top author* yang diambil dari *Google Scholar*,dan *Excellence* merupakan jumlah artikel-artikel ilmiah yang publikasikan oleh Perguruan Tinggi yang bersangkutan dan terindeks di *Scimago.*

## SEMRushRank

SEMRush memiliki sistem perangking yaitu SEMRush Rank, yang mana merupakan suatu jenis alat *Search Engine Optimized* (SEO) yang digunakan untuk melacak rangking kata kunci pada mesin pencari. SEMrush Rank memberikan kemudahan kepada pemilik situs web untuk melihat rangking pada kata kunci yang akan digunakan dan jumlah berapa kali kata kunci tersebut digunakan.

Alasan kenapa *keyword* sangat penting adalah karena setiap pengguna internet akan mengetikan kata dalam search engine sesuai dengan kebutuhannya dan *keyword* inilah yang membuat suatu *website* dikunjungi (Rahayu, 2018). Macam-macam *keyword* antara lain:

## Long Tail dan ShortTail

*Long tail* atau adalah *Short tail* permisalan yang menuju pada penggunaan katakunci. Arti dari *longtail* adalah kata kunci yang terdiri lebih dari dua kata, sedangkan *short tail* adalah, kata kunci yang hanya terdiri dari satu sampai dua baris kata saja.

## Brand Keyword dan GenericKeyword

Istilah *Brand Keyword* dan *Generic Keyword* menuju pada kata kunci yang berkaitan erat dengan suatu *website*, seperti tema, isi, dan gaya penulisan yang dipakai oleh suatu *website*.

## GoogleIndex

Pada artikel *Search Console* disitu menjelaskan bahwa isi dari pelaporan status index (Google Index) menampilkan jumlah total URL yang tersedia untuk ditampilkan dihasil penelusuran beserta URL lain yang dapat ditemukan oleh Google dengan cara lain, jumlah ini berubah dari waktu ke waktu saat pengguna menambah dan menghapus halaman.

## Domain

Menurut (Afif, 2014) umur dari sebuah domain tidak mengacu dari berapa lama suatu *website* telah memiliki nama domain, melainkan berapa lama Google mengindeks domain tersebut. Google tidak mengindeks halaman *website*secaraacak,umur dari domain dan *sitei ndex* yang lebih banyak menjadi prioritas utama Google untuk menentukan kualitas sebuah *website*.

## Backlinks

Backlink adalah link yang menghubungkan dua halaman *website* (Ika Nur Solechah, 2015). Penggunaan Backlink pada *website* itu penting selain untuk menghubungkan antar halaman, backlink juga dapat dikatakan sebagai tolok ukur kualitas konten dari suatu *website*. Menurut (Febry Eka Purwiantono, 2014) semakin besar jumlah *backlink* suatu situs, maka semakin populer dan penting keberadaan maupun konten dari situs tersebut.

## Search Engine Optimization (SEO)

Optimisasi Mesin Pencari atau Search Engine Optimization (SEO) adalah sebuah proses untuk mempengaruhi tingkat keterlihatan (visibilitas) sebuah situs web atau sebuah halaman web dihasil pencarian alami (sering disebut juga dengan pencarian tak-berbayar, pencarian non-iklan, atau pencarian organik) dari sebuah mesin pencari (Ortiz-Cordova, A. and Jansen, B. J.,2012).

Tujuan dari SEO adalah agar web atau blog selalu berada di halaman terdepan dan teratas dari suatu Search Engine, sehingga selalu berada di posisi tearatas, maka besar kemungkinan web atau blog sering dikunjungi.

Secara garis besar, teknik yang digunakan dalam penerapan metode SEO ada dua, yaitu:

1. SEO On-Page (Gambar 2.4) Merupakan sebuah teknik SEO yang difokuskan untuk optimasi pada internal *website*. Aktifitas On-Page ini menyangkut rekayasa terhadap elemen dan isi sebuah *website*.



**Gambar 2.4 Bagan SEO On-Page**

1. SEO Off-Page (Gambar 2.5) merupakan langkah optimasi SEO terhadap suatu *website* yang dilakukan diluar halaman (eksternal) *website* tersebut. Metodeoptimasi Off-Page sangat erat kaitannya dengan backlink. Backlink adalah link dari *website* lain ke *website* yang dimaksud. Sebuah backlink dapat diibaratkakan sebuah rekomendasi, semakin banyak yang merekomendasikan sebuah *website* maka akan semakin besar pula *website* tersebut dalam mendapatkan perhatian dari mesin pencari.



**Gambar 2.5 Bagan SEO Off-Page**

## Flowchart

Menurut Munir (2016) Bagan alir (flowchart) adalah bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.Padawaktuakanmenggambarsuatubaganalir, analis sistem atau pemrogam dapat mengikuti pedoman-pedoman sebagai berikutini.

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatuhalaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan denganjelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatupekerjaan.
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbolpenghubung.
7. Gunakanlah simbol-simbol bagan alir yangstandar.

Menurut Munir (2016) Ada lima macam bagan alir yang akan dibahas di modul ini, yaitu sebagai berikut ini.

* 1. Bagan alir sistem (*SystemsFlowchart*).
	2. Bagan alir dokumen (*DocumentFlowchart*).
	3. Bagan alir skematik (*SchematicFlowchart*).
	4. Bagan alir program (*ProgramFlowchart*).
	5. Bagan alir proses (*ProcessFlowchart*).

## Pengertian Unified Modeling Language (UML)

Pengertian Sistem Menurut Nugroho (2014:105): “UML (Unified Modelling Language) adalah bahasa untuk menspesifikasikan,

memvisualisasikan, serta mengkontruksi bangunan dasar dari sistem perangkat lunak termasuk didalamnya dengan melibatkan pemodelan aturan-aturan bisnis. Macam diagram UML antara lain *Use Case Diagram, Class Diagram, Activity Diagram, Squense Diagram*.

*Use Case Diagram* (Tabel 2.1) memiliki simbol-simbol dan arti nya. Setiap simbol memiliki fungsi yang berbeda karena itu setiap programer wajib mengetahui demi mempermudah rancangan perangkat lunak yang akan dibuat.

**Tabel 2.1 Notasi *Use Case Diagram***

|  |  |  |
| --- | --- | --- |
| **SIMBOL** | **NAMA** | **KETERANGAN** |
|  | ***Actor*** | *Actor* adalah pengguna sistem. *Actor* tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut jugabisa dianggap sebagai actor. |
|  | ***Use Case*** | *Use case* digambarkan sebagai lingkaran elips dengan nama use case dituliskan didalamelips tersebut. |
|  | ***Association*** | Asosiasi digunakan untukmenghubungkan actor dengan |

|  |  |  |
| --- | --- | --- |
|  |  | use case. Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara *Actor*dengan *Use Case*. |

Salah satu dari diagram - diagram yang ada pada UML, *Sequence Diagram* (Tabel2.2)ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara object.

**Tabel 2.2 Notasi *Sequence Diagram***

|  |  |  |
| --- | --- | --- |
| **SIMBOL** | **NAMA** | **KETERANGAN** |
|  | **Object** | Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyekdidalamnya yang diawali dengan sebuah titik koma |
|  | **Actor** | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |
|  | **Lifeline** | Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek. |

|  |  |  |
| --- | --- | --- |
|  | **Activation** | Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi. |
|  | **Message** | Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object. |

*Activity diagram* (Tabel 2.3), sesuai dengan namanya diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat.

**Tabel 2.3 Notasi *Activity Diagram***

|  |  |
| --- | --- |
| **SIMBOL** | **KETERANGAN** |
|  | Titik Awal |
|  | Titik Akhir |
|  | Activity |
|  | Pilihan Untuk mengambil Keputusan |

*Class Diagram* (Tabel 2.4) adalah model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class.Class diagram mirip

ER-Diagram pada perancangan database, bedanya pada ER-diagram tidak terdapat operasi/methode tapi hanya atribut.

**Tabel 2.4 Notasi *Class Diagram***

|  |  |  |
| --- | --- | --- |
| **SIMBOL** | **NAMA** | **KETERANGAN** |
|  | **Assosiation** | Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 class, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*.Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum- hukum multiplisitas pada sebuah *relationship* (Contoh: One-to-one, one-to-many, many-to-many). |
|  | **Composition** | Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari*class* yang lain, maka *class* |

|  |  |  |
| --- | --- | --- |
|  |  | tersebut memiliki relasi *Composition* terhadap *class* tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentukjajaran genjang berisi/solid. |
|  | **Dependency** | Kadangkala sebuah *class* menggunakan *class* yang lain. Hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagaisebuah panah bertitik-titik. |
|  | **Aggregation** | *Aggregation*mengindikasikan |

|  |  |  |
| --- | --- | --- |
|  |  | keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah *aggregation* digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidakberisi/tidak solid. |
|  | **Generalization** | Sebuah relasi *generalization* sepadan dengan sebuah relasi *inheritance* pada konsep berorientasi obyek. Sebuah *generalization* dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “*paren*t”-nya/induknya. |

## MySql

MySQL adalah database server relasional yang gratis dibawah lisensi GNU *General Public License*. Dengan sifatnya yang Open Source, memungkinkan user juga untuk melakukan modifikasi pada *source code*-nya untuk memenuhi kebutuhan spesifik mereka sendiri.MySQL merupakan *database* server multi - user dan multi – threded yang tangguh (robust). Dengan memiliki banyak feature MySQL bisa bersaing dengan database komersial sekalipun. Tidak mengejutkan, MySQL menjadi database pilihan untuk banyak pengguna PHP.

MySQL dikembangkan oleh MySQL AB, sebuah perusahaan komersial yang membangun layanan bisnisnya melalui database MySQL. Awal mula pengembangan MySQL adalah penggunaan MySQL untuk koneksi ke *table* mempergunakan rutin level rendah (ISAM) setelah beberapa pengujian diperoleh kesimpulan mSQL tidak cukup cepat dan fleksibel untuk memenuhi kebutuhan.Sehingga dihasilkan suatu antar muka SQL baru pada database tetapi dengan API yang mirip mSQL. API ini dipilih sedemikian rupa sehingga memudahkan porting kode. Tentang penggunaan nama MySQL sampai saat ini belum jelas betul. Base directory dan sejumlah library serta tool pada saat pengembangan telah diawali dengan “my” selama sekitar 10 tahun. Bagaimanapun pemberia nnama MySQL sampai saat ini masih menjadi sebuah “misteri”.

MySQL adalah suatu sistem manajemen database. Untuk menambahkan, mengakses, dan memproses data yang tersimpan pada suatu *database* komputer

yang memerlukan sistem manajemen database seperti MySQL. Karena komputer sangat unggul dalam menangani sejumlah besar data, system manajemen database memainkan suatu peranan yang penting dalam komputansi, baik sebagai utility stand-alone maupun bagian dari aplikasi lainnya.

MySQL adalah sistem manajemen database relasional. Suatu database relasional menyimpan data dalam tabel-tabel terpisah. Hal ini memungkin kecepatan dan fleksibelitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa table pada suatu permintaan. Bagian SQL dari kata MySQL berasal dari “*StructuredQuery Language*” bahasa paling umum yang dipergunakan untuk mengakses database. Konektivitas, kecepatan dan keamanannya membuat MySQL cocok untuk pengaksesan database pada internet. MySQL merupakan sistem client/server yang terdiri dari SQL server multiheraded yang memungkinkan *backed* yang berbeda, sejumlah program client dan library yang berbeda, tool *administrative*, dan beberapa antarmuka pemrograman. MySQL juga tersedia sebagai *library* yang bisa digabungkan keaplikasi.

Karena pada MySQL tersedia banyak perangkat lunak yang ikut berkontribusi, maka pengguna bisa mendapatkan aplikasi atau bahasa favorit yang sudah didukung oleh MySQL.Cara mengeja MySQL adalah“*MyEssQue Ell*” (bukan My-Squell).

## Application Programming Interface (API)

*Application Programming Interface (API)* merupakan satu set instruksi pemograman untuk mengakses aplikasi berbasis web software atau web tool. Sebuah perusahaan perangkat lunak merilis *API* kepada publik sehingga pengembang perangkat lunak lain dapat merancang produk yang didukung oleh layanannya (Dea Chintia, 2012). *Application Programming Interface (API)* pun dapat didefinisikan sebagai sebuah bahasa yang digunakan oleh sebuah program aplikasi untuk berkomunikasi dengan sistem operasi atau program kontrol lainnya seperti *Database Management Sistem (DBMS)* atau protokol komunikasi.

Sebagai contoh: *google.com* merilis *Application Programming Interface (API)* sehingga pengembang situs web dapat dengan mudah mengakses informasi Pagerank dari seluruh web/domain di dunia. *Application Programming Interface (API)* adalah antar muka *software to software*, bukan *user interface*. Dengan *Application Programming Interface (API)*, aplikasi berkomunikasi satu sama lain tanpa sepengetahuan.Contohnya ketika seseorang membeli tiket film secara online dan memasukkan informasi kartu kredit, situs tiket film menggunakan *Application Programming Interface (API)* untuk mengirimkan informasi kartu kredit ke remote aplikasi yang memverifikasi apakah informasi sudah benar. Setelah pembayaran dikonfirmasi, *remote* aplikasi mengirimkan respon balik ke situs tiket film dan mengatakan pembayaran telah diterima dan tiket dapat diberikan. Sebagai pengguna hanya bisa melihat satu antarmuka yaitu situs web tiket film. Tetapi dibelakang layar

banyak aplikasi yang bekerja bersama-sama menggunakan *Application Programming Interface (API)*.

## CodeIgniter

Menurut Hakim (2010:8) CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal.



**Gambar 2.6 Logo CodeIgniter**

CodeIgniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. [(http://ellislab.com)](http://ellislab.com/), sebuah perusahaan yang memproduksi CMS (*Content*

*Management Sistem*) yang cukup handal, yaitu *Expression Engine*

[(http://www.expressionengine.com)](http://www.expressionengine.com/). Saat ini, CodeIgniter dikembangkan dan

dimaintain oleh *Expression Engine Development Team.*

Adapun beberapa keuntungan menggunakan CodeIgniter, diantaranya:

1. Gratis

CodeIgniter berlisensi dibawah Apache/BSD opensorce.

1. Ditulis Menggunakan PHP4

Meskipun CodeIgniter dapat berjalan di PHP 5, namun sampai saat ini kode program CodeIgniter masih dibuat dengan menggunakan PHP 4.

1. Berukuran Kecil

Ukuran CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

1. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkanpemisahan

*layer application-logic* dan *presentation.*

1. URL yang Sederhana

Secara default, URL yang dihasilkan CodeIgniter sangat bersih dan *Serach Engine Friendly* (SEF).

1. Memiliki Paket *Library* yang Lengkap

CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasioperasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses *database*, mengirim email, memvalidasi form, menangani *session* dan sebagainya.

1. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan

*helper,* atau dengan menggunakan *hooks.*

1. Tidak Memerlukan *Template Engine*

Meskipun CodeIgniter dilengkapi dengan *template* parser sederhanayang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

1. Dokumentasi Lengkap dan Jelas

Dari sekian banyak *framework*, CodeIgniter adalah satu-satunya*framework* dengan dokumentasi yang lengkap dan jelas.

1. Komunitas

Komunitas CodeIgniter saat ini berkembang pesat. Salah satukomunitasnya bisa dilihat di ([http://codeigniter.com/forum/)](http://codeigniter.com/forum/).