BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Umum Perusahaan

2.1.1 Sejarah Perusahaan

BIMS (Bima Inspira Master Solution) – merupakan korporasi yang bergerak dibidang jasa konsultasi berbagai kepentingan dibidang Sistem Informasi dan Teknologi. BIMS menitik beratkan layanan pada : Pembuatan dan pengembangan aplikasi/sistem berbasis Web/Mobile yang tepat sasaran pada problem milik client dan berorientasi pada solusi.

Kebidangan Utama :

1.Geographical Information System (GIS)

2. Android Mobile Aplication

3. Data WareHouse (ETL Designer)

Terminologi BIMS:

Bima adalah anak ke 2 dari pandawa lima, yang paling kuat, kasar tapi berhati lembut, harapan kami -> BIMS dapat menjadi korporasi kuat “valuable” layaknya bima, dengan segala birokrasi dan harapan bisnis didalamnya tapi tepat mengedepankan kepentingan umat didalamnya.

Inspira -> merepresentasikan objek yang menginspirasi, harapan kami -> BIMS dapat menjadi konsultan IT yang menginspirasi berbagai kalangan mulai dari rekan – rekan sesama konsultan, client kami, serta berbagai pihak terhubung lainnya.

Master -> merepresentasikan alhi/tukang, harapan kami BIMS sebagai pihak yang memang ahli dibidangnya, dengan bantuan tangan – tangan cekatan para pemikir didalamnya BIMS membangun solusi untuk lingkungan.

Solution -> merepresentasikan BIMS sebagai lembagi yang solutif, bukan lembaga yang seakan hanya menambah masalah baru para pengguna jasanya.

Logo Merah dan Putih melambangkan Bangsa Indonesia, tulisan jelas disebelah kanan makna “To The Point”, dengan flat desain kepala wayang yang merepresentasikan Bima,

“Free hands design” pada bagian kepala wayang merepresentasikan “kreativitas”.

Berawal dari perjalan hobby, bisnis dan keinginan berkontribusi untuk negeri, kami alumni POLBAN (Politeknik Negeri Bandung) Jurusan Teknik Komputer dan Informatika berkumpul, bernegosiasi, menyamakan visi kerja untuk membangun suatu korporasi profesional dalam rangka beraktivitas , bekerja kreatif, inovatif untuk membantu lingkungan dan berkontribusi aktif dalam perkembangan bangsa.

2.1.2 Tempat dan Kedudukan Perusahaan

Lokasi yang menjadi tempat kerja praktek penulis berada di Jalan Gg. Sarirasa 5 No.41 Sarijadi Blok 9 Kelurahan Sukasari, Kecamatan Sukajadi, Kota Bandung 40151

2.1.3 Logo Perusahaan



Gambar 2.2 Logo Perusahaan

2.1.4 Bentuk dan Badan Hukum Perusahaan

Korporasi BIMS secara legal di dirikan pada 2 Oktober 2016 sebagai CV yang bergerak dibidang konsultasi berbagai hal terkait Sistem Informasi dan Teknologi, dan berbagai hal yang berhubungan dengannya.

2.2 Aplikasi

Aplikasi berasal dari kata application yaitu bentuk benda dari kata kerja to apply yang dalam bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Contoh utama perangkat lunak aplikasi adalah program pengolah kata, lembar kerja, dan pemutar media. Kumpulan aplikasi komputer yang digabung menjadi suatu paket biasanya disebut paket atausuite aplikasi (application suite). Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Umumnya aplikasi-aplikasi tersebut memiliki kemampuan untuk saling berinteraksi sehingga menguntungkan pemakai. Contohnya, suatu lembar kerja dapat dimasukkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan printah-printah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

Pengertian aplikasi menurut para ahli adalah sebagai berikut :

1. Menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

2. Menurut Kamus Kamus Besar Bahasa Indonesia (1998 : 52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atauketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

3. Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows &, permainan (game), dan sebagainya.

4. Menurut Harip Santoso, Aplikasi adalah suatu kelompok file (form, class, rePort) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi fixed asset, dan lain-lain.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan OpenOffice.org, Bahasa Pemrograman yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi

Berdasarkan jenisnya, aplikasi komputer dapat dibagi menjadi beberapa kategori, yaitu :

1. Enterprise

Digunakan untuk organisasi yang cukup besar dengan maksud menghubungkan aliran data dan kebutuhan informasi antar bagian, contoh : IT Helpdesk, Travel Management dan lain-lain.

2. Enterprise – SupPort

   Sebagai aplikasi pendukung dari Enterprise, contohnya : Database Management, Email Server dan Networking System.

3. Individual Worker

   Sebagai aplikasi yang biasa digunakan untuk mengolah/edit data oleh tiap individu. Contoh : Ms.Office, Photoshop, Acrobat Reader dan lain-lain.

4. Aplikasi Akses Konten

   Adalah aplikasi yang digunakan oleh individu (hanya) untuk mengakses konten tanpa kemampuan untuk mengolah atau mengedit datanya melainkan hanya melakukan kustomisasi terbatas. Contoh : Games, Media Player, Web Browser.

5. Aplikasi Pendidikan

   Biasanya berbentuk simulasi dan mengandung konten yang spesifik untuk pembelajaran.

6. Aplikasi Simulasi

   Biasa digunakan untuk melakukan simulasi penelitian, pengembangan dan lain-lain. Contoh : Simulasi pengaturan lampu lalu lintas.

7. Aplikasi Pengembangan Media

   Berfungsi untuk mengolah/mengembangkan media biasanya untuk kepentingan komersial, hiburan dan pendidikan. Contoh : Digital Animation Software, AudioVideo Converter dan lain-lain.

8. Aplikasi Mekanika dan Produk

Dibuat sebagai pelaksana/pengolah data yang spesifik untuk kebutuhan tertentu. Contoh : Computer Aided Design (CAD), Computer Aided Engineering (CAE), SPSS dan lain-lain.

2.3 Task

Menurut KBBI Task atau tugas adalah sesuatu pekerjaan yang harus dikerjakan atau dieselesaikan. Akan tetapi task atau tugas yang dimaksud dalam tugas akhir ini adalah Bagian terkecil dari pekerjaan yang dapat dikenali dan berfungsi sebagai unit kerja.

2.4 Management

Definisi Manajemen adalah suatu seni mengarahkan orang lain untuk mencapai tujuan utama dalam suatu organisasi melalui proses perencanaan (Planning), pengorganisasian (Organizing), dan mengelola (Controlling) sumber daya manusia dengan cara efektif dan efisien.

Pengertian dan Definisi Manajemen menurut Para Ahli

1. Manajemen adalah adalah suatu proses yang berbeda terdiri dari planning, organizing, actuating, dan controlling yang dilakukan untuk mencapai tujuan yang ditentukan dengan menggunakan manusia dan sumber daya lainnya (George R. Terry, 1997).

2. Manajemen adalah suatu seni yang produktif yang didasarkan pada suatu pemahaman ilmu, ilmu dan seni tidaklah bertentangan, namun masing masing saling melengkapi (Koontz)

3. Ilmu Manajemen merupakan proses dalam membuat suatu perencanaan, pengorganisisasian, pengendalian serta memimpin berbagai usaha dari anggota entitas atau organisasi dan juga mempergunakan semua sumber daya yang dimiliki untuk mencapai tujuan yang ditetapkan (Stoner)

4. Manajemen sebagai sebuah rangkaian tindakan tindakan yang dilakukan oleh para anggota organisasi dalam upaya mencapai sasaran organisasi.

prosess merupakan suatu rangkaian aktivitas yang dijalankan dengan sistematis (Wilson)

5. Manajemen adalah sebuah seni dalam mencapai tujuan yang diinginkan yang dilaksanakan dengan usaha orang yang lain (Lawrance A Appley)

6. Manajemen sebagai suatu seni, tiap tiap pekerjaan bisa diselesaikan dengan orang lain (Mary Parker F)

2.5 HTML

HTML (*Hyper Text Markup Language*) digunakan untuk menampilkan konten pada *browser*. Konten HTML berupa tulisan, gambar maupun tabel. Dalam menampilkan konten pada *browser*, HTML menggunakan kode yang biasa disebut *tag*.

Beberapa *tag* yang umum digunakan dalam pembuatan atau pengembangan aplikasi adalah:

1. Html : File HTML sendiri dimulai dengan tag <html> dan diakhiri oleh </html>.

2. Table : Untuk membuat table di HTML, diawali dengan tag <table> lalu untuk membuat baris digunakan tag <tr> dan untuk membuat kolom menggunakan tag <td>

3. p : Tag <p> digunakan dalam pembuatan teks paragraph.

4. b, i, u : Tag ini digunakan untuk membuat gaya tulisan seperti bold, italic dan underline

5. a : Tag <a> digunakan untuk membuat link menuju halaman website lain.

6. Img : Tag <img> digunakan untuk menampilkan gambar pada halaman website.

2.6 PHP

PHP adalah bahasa pemrograman server yang sangat dinamis dan interaktif, yang sangat ampuh untuk membuat sebuah halaman web. Bahasa pemograman PHP digunakan secara gratis dan bebas, tentunya hal ini adalah alternatif yang sangat bagus dan efisien di banding jika kita menggunakan pesaing seperti Microsoft ASP yang berbayar.

PHP adalah singkatan dari PHP : Hypertext Preprocessor. Di mana huruf ‘P’ pada kata PHP merupakan singkatan dari kata PHP itu sendiri atau yang sering disebut rekursif.

Maksud dari rekursif adalah PHP memiliki kemampuan / fungsi untuk memanggil dirinya sendiri. Pengertian PHP adalah bahasa pemrograman script server side yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web.

PHP merupakan bahasa pemrograman paling populer yang digunakan dalam membuat sebuah website. Berdasarkan survei dari Wtechs, bahasa pemrograman PHP merupakan bahasa pemrograman server-side yang paling populer dibandingkan ASP.NET maupun dengan Java.

Pada dasarnya bahasa pemrograman web adalah HTML, namun untuk membuat web menjadi dinamis sangat di perlukan bahasa pemrograman PHP, maka dari itu penggunakan PHP adalah ditanamkan atau disisipkan di dalam sebuah pemrograman HTML.



Gambar 2.3 Logo php

2.7 MySQL

MySQL adalah perangkat lunak open source yang digunakan untuk manajemen basis data atau bias disebut *Database Management System* (DMBS). MySQL banyak digunakan sebagai database pada aplikasi berbasis web.

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, dan lain-lain.

2. *Open Source*. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Cepat. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tupe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

7. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar barus. Selain itu batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.

Beberapa contoh query atau perintah yang digunakan pada MySQL ialah sebagai berikut:

1. CREATE DATABASE

```
CREATE (DATABASE | SCHEMA)  (IF NOT EXISTS) db_name
      [create_specification] …
```

Query diatas digunakan untuk membuat sebuah *database* baru.

2. CREATE TABLE

```
Create [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
(create_definition,…)
[table_oprions]
[partition_options]
```

Query diatas digunakan untuk membuat sebuah tabel baru beserta kolom-kolom yang akan dibuat didalamnya.

3. DROP TABLE

```
DROP [TEMPORARY] TABLE [IF EXISTS]
      Tbl_name [, tbl_name] …
```

Query diatas digunakan untuk menghapus satu atau lebih tabel yang ada di *database*. Seluruh struktur tabel dan data yang ada pada tabel tersebut akan terhapus.

4. RENAME TABLE

```
RENAME TABLE tbl_name TO new_tbl_name
      [, tbl_name2 TO new_tbl_name2] …
```

Query diatas digunakan untuk mengganti nama tabel yang ada di *database*.

5. SELECT

```
Select [ALL | DISTINCT | DISTINCTROW ] [STRAIGHT_JOIN]
      Select_expr [, select_expr …] [FROM table_references]
      [WHERE where_condition] [GROUP BY {col_name | expr |
position} [ASC | DESC], … [WITH ROLLUP]]
      [HAVING where_condition] [ORDER BY {col_name | expr |
position} [ASC | DESC] , …] [LIMIT {[offset,] row_count |
row_count OFFSET offset}]
```

Query diatas digunakan untuk mendapatkan data dari satu atau lebih tabel yang ditentukan. Kode select_expr digunakan untuk mem-*filter* kolom mana saja yang akan ditampilkan datanya, sedangkan kode where_condition digunakan untuk menyatakan kondisi pada setiap baris data yang harus terpenuhi untuk ditampilkan datanya.

6. INSERT

```
INSERT [INTO] tbl_name
      [PARTITION (partition_name,…)]
      [(col_name,…)]
      {VALUES | VALUE} ({expr | DEFAULT},…),{…},…)
```

Query diatas digunakan untuk menambahkan baris data pada tabel yang ditentukan.

7. Update

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
      SET            [col_name1={expr1|DEFAULT}            {,
col_name2={expr2|DEFAULT}] …
      [WHERE where_condition]
      [ORDER BY …]
```

```
[LIMIT row_count]
```

Query diatas digunakan untuk merubah nilai dari baris data yang ada pada tabel yang ditentukan dengan nilai yang baru.

8. DELETE

```
DELETE [LOW_PRIORITY] [QUICE] [IGNORE] FROM tbl_name
[PARTITION (partition_name,…)]
[WHERE where_condition]
[ORDER BY …]
[LIMIT row_count]
```

Query diatas digunakan untuk menghapus baris data pada tabel yang ditentukan.

2.8 WEB

Situs web adalah suatu halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti Internet, ataupun jaringan wilayah lokal (LAN) melalui alamat Internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di Internet disebut pula sebagai World Wide Web atau lebih dikenal dengan singkatan WWW. Meskipun setidaknya halaman beranda situs Internet umumnya dapat diakses publik secara bebas, pada praktiknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa situs web mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota, atau bahkan meminta pembayaran untuk dapat menjadi aggota untuk dapat mengakses isi yang terdapat dalam situs web tersebut, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surel (e-mail), dan lain-lain. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi, atau karena tujuan komersial tertentu.

Sebuah halaman web merupakan berkas yang ditulis sebagai berkas teks biasa (plain text) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML atau XHTML, kadang-kadang pula disisipi

dengan sekelumit bahasa skrip. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer.

Halaman-halaman web tersebut diakses oleh pengguna melalui protokol komunikasi jaringan yang disebut sebagai HTTP, sebagai tambahan untuk meningkatkan aspek keamanan dan aspek privasi yang lebih baik, situs web dapat pula mengimplementasikan mekanisme pengaksesan melalui protokol HTTPS

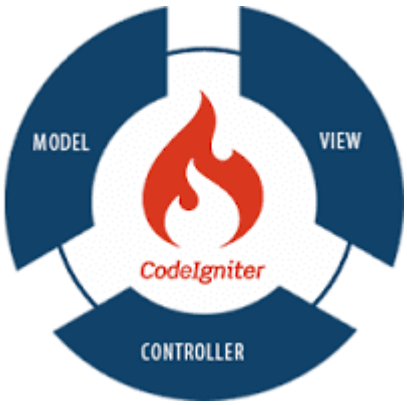
2.9 Code Igniter

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis. CodeIgniter pertama kali ditulis oleh Rick Ellis (http://www.ellislab.com), seorang musisi rock yang beralih profesi menjadi programmer dalam riset kecil-kecilannya dan menghasilkan suatu framework PHP yang berukuran kecil, ringan serta memenuhi fitur umum aplikasi PHP. Namun, sejak tahun 2014 CodeIgniter telah dimiliki oleh British Columbia Institute of Technology (BCIT).

CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap menjadi salah satu alasan kuat banyak orang memilih CodeIgniter sebagai framework. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena “it is faster, lighter and the least like a framework.”

Disamping itu, penggunaan CodeIgniter juga akan menghasilkan suatu struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya. Struktur aplikasi yang rapi tentu sangat diperlukan dari sebuah aplikasi. Misalnya, jika terjadi suatu error dalam aplikasi, dengan code yang rapi kita dapat dengan mudah menemukan kesalahan tersebut. Bukan hanya itu, bayangkan suatu saat

aplikasi yang kita bangun membutuhkan fungsi-fungsi lain yang sangat penting, tentu akan diperlukan pengembangan lebih lanjut. Dan sekali lagi, hal itu juga sangat memerlukan struktur coding yang rapi. Dengan menggunakan CodeIgniter, untuk mewujudkan struktur kode yang rapi sangat-sangat mungkin terjadi. Hal itu dikarenakan CodeIgniter dibangun berbasis **MVC** (Model, View, Controller) yang memisahkan antara tampilan dan logic aplikasi.



Gambar 2.4 MVC Code Igniter

1. **Model** adalah bagian yang bertanggung jawab terhadap operasi database, baik itu create, read, update atau delete. Berisi berupa fungsi-fungsi operasional database yang dapat dipanggil oleh Controller.

2. **View** adalah bagian yang menangani tampilan. bagian inilah yang bertugas untuk mempresentasikan data kepada user. Ia berbentuk struktur HTML yang berisikan variabel data yang dikirimkan oleh Controller.

3. **Controller** adalah bagian yang mengatur hubungan antara Model dan View. Ia adalah otak dari kinerja aplikasi. Ia terdiri dari fungsi-fungsi yang bersifat operasional dan logikal. Saat ada request yang masuk, ia akan menangani dan memprosesnya untuk kemudian ditampilkan dalam View.

Selain itu terdapat kelebihan dan kekurangan dari Code Igniter. Berikut merupakan kelebihan dari CodeIgniter.

1. Berukuran sangat kecil. File download nya hanya sekitar 2MB, itupun sudah include dokumentasinya yang sangat lengkap.
2. Dokumentasi yang bagus. Saat anda mendownloadnya, telah disertakan dengan dokumentasi yang berisi pengantar, tutorial, bagaimana panduan penggunaan, serta referensi dokumentasi untuk komponen-komponennya.
3. Kompitabilitas dengan Hosting. CodeIgniter mampu berjalan dengan baik pada hampir semua platfom hosting. CodeIgniter juga mendukung database-database paling umum, termasuk MySQL.
4. Tidak ada aturan coding yang ketat. Terserah anda jika anda hanya ingin menggunakan Controller, tanpa View, atau tidak menggunakan Model, atau tidak salah satu keduanya. Namun dengan menggunakan ketiga komponennya adalah pilihan lebih bijak.
5. Kinerja yang baik. Codeigniter sangat cepat bahkan mungkin bisa dibilang merupakan framework yang paling cepat yang ada saat ini.
6. Sangat mudah diintegrasikan. CodeIgniter sangat mengerti tentang pengembangan berbagai library saat ini. Karenanya CodeIgniter memberikan kemudahan untuk diintegrasikan dengan library-library yang tersedia saat ini.
7. Sedikit Konfigurasi. Konfigurasi CodeIgniter terletak di folder aplication/config. CodeIgniter tidak membutuhkan konfigurasi yang rumit, bahkan untuk mencoba menjalankannya, tanpa melakukan konfigurasi sedikitpun ia sudah bisa berjalan.
8. Mudah dipelajari. Disamping dokementasi yang lengkap, ia juga memiliki berbagai forum diskusi.

Berikut merupakan kekurangan CodeIgniter

1. Library yang sangat terbatas. Hal ini dikarenakan sangat sulit mencari plugin tambahan yang terverifikasi secara resmi, karena pada situsnya CodeIgniter tidak menyediakan plugin-plugin tambahan untuk mendukung pengembangan aplikasi dengan CI.
2. Belum adanya editor khusus CodeIgniter, sehingga dalam melakukan create project dan modul-modulnya harus berpindah-pindah folder.

2.10 *Object Oriented Programming* (OOP)

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

Berikut merupakan karakteristik OOP (Object Oriented Programming) :

- Semua adalah objek.
- Komputasi dilakukan dengan komunikasi antar objek. Setiap objek berkomunikasi dengan objek yang lain melalui pengiriman dan penerimaan pesan.
- Sebuah pesan merupakan permintaan atas sekumpulan aksi dengan semua argumen yang diperlukan untuk menyelesaikan suatu tugas tertentu.

- Setiap objek memiliki memori sendiri, yang dapat terdiri dari objek-objek lainnya.
- Setiap objek adalah wakil atau representasi dari suatu kelas. Sebuah kelas dapat mewakili sekelompok objek yang sama.
- Kelas merupakan kumpulan tingkah laku yang berkaitan dengan suatu objek. Jadi, semua objek yang merupakan wakil dari kelas yang sama dapat melakukan aksi yang sama pula.
- Kelas-kelas diorganisasikan ke dalam struktur pohon yang berakar tunggal, yang dinamakan dengan jenjang pewarisan (inheritance hierarchy).
- Setiap objek pada umumnya memiliki tiga sifat, yaitu keadaan, operasi dan identitas objek.
- Operasi merupakan tindakan yang dapat dilakukan oleh sebuah objek.
- Keadaan objek merupakan koleksi dari seluruh informasi yang dimiliki oleh objek pada suatu saat.
- *Informasi yang terkandung pada objek tersebut pada akhirnya memberikan identitas khusus yang membedakan suatu objek dengan objek lainnya.*

Berikut merupakan konsep OOP (Object Oriented Programming) :

1. **Enkapsulasi (encapsulation)**
    - Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
    - Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
    - Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.

- Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).

2. **Enkapsulasi (encapsulation)**

   - Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
   - Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
   - Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
   - Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

3. **Pewarisan (Inheritance)**

   - Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
   - Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
   - Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
   - Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.

4. **Polimorfisme (polymorphism)**

- Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
- Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
- Method overriding.
- Method name overloading.

2.11 Unified Modeling Languange (UML)

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML.

UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal:

- Object
- Class
- Abstraction

- Encapsulation
- Inheritance
- Polymorphism

Dalam UML sendiri terdapat beberapa diagram yang wajib dikuasai yaitu:

- Structural Diagram
- Class Diagram, diagram ini terdiri dari *class*, *interface*, *association*, dan *collaboration*. Diagram ini menggambarkan objek - objek yang ada di sistem.
- Object Diagram, diagram ini menggambarkan hasil instansi dari *class diagram*. Diagram ini digunakan untuk membuat *prototype*
- Component Diagram, diagram ini menggambarkan kumpulan komponen dan hubungan antar komponen. Komponen terdiri dari *class*, *interface*, atau *collaboration*
- Deployment Diagram, diagram ini menggambarkan kumpulan *node* dan hubungan antar *node*. *Node* adalah entitas fisik dimana komponen di-*deploy*. Entitas fisik ini dapat berupa *server* atau perangkat keras lainnya.
- Behavioral Diagram
- Use case Diagram, diagram ini menggambarkan kumpulan *use case*, aktor, dan hubungan mereka. *Use case* adalah hubungan antara fungsionalitas sistem dengan aktor internal/eksternal dari sistem.
- Sequence Diagram, diagram ini menggambarkan interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya.
- Collaboration Diagram, diagram ini merupakan bentuk lain dari *sequence diagram*. Diagram ini menggambarkan struktur organisasi dari sistem dengan pesan yang diterima dan dikirim.

- Statechart Diagram, diagram ini menggambarkan bagaimana sistem dapat bereaksi terhadap suatu kejadian dari dalam atau luar. Kejadian (*event*) ini bertanggung jawab terhadap perubahan keadaan sistem.
- Activity Diagram, menggambarkan aliran kontrol sistem. Diagram ini digunakan untuk melihat bagaimana sistem bekerja ketika dieksekusi.

UML umum digunakan di dunia kerja dan dunia pendidikan. Dunia kerja menggunakan UML untuk menggambarkan sistem dan aplikasi berbasis objek yang sudah ada sebelumnya atau merancang sistem dan aplikasi di awal pengembangan sebelum memasuki tahap *coding*. Beberapa posisi pekerjaan yang sering bersinggungan dengan UML antara lain *system analyst*, *software architect*, dan *technical writer*. *System analyst* selain perlu memiliki kemampuan *coding*, sudah tentu wajib menguasai UML, jika akan mengembangkan aplikasi atau sistem berbasis objek agar dapat menjadi acuan formal bagi para *developer* yang bekerja sama dengannya. Tidak hanya itu *system analyst* pun terkadang dituntut untuk menguasai pembuatan diagram lain seperti *data flow diagram*, *flowchart*, *entity relationship diagram*, dan lainnya. Sedangkan UML menjadi kemampuan wajib bagi seorang *technical writer*. Pekerjaan di posisi ini cukup banyak dibutuhkan oleh beberapa perusahaan yang kesulitan dalam membuat dokumentasi sistem atau aplikasi yang mereka kembangkan. Selain membuat *user guide*, *technical writer*diperlukan untuk membangun keseluruhan dokumentasi dalam UML dari aplikasi atau sistem, sehingga keberadaannya dapat menambah nilai aplikasi menjadi lebih tinggi.

UML pun digunakan sebagai alat dalam beberapa mata kuliah di perguruan tinggi yang membuka jurusan ilmu komputer, teknik informatika, sistem informasi, manajemen informatika, dan komputerisasi akuntansi. Mata kuliah yang biasa diberikan bernama Metodologi Berorientasi Objek dan Pemrograman Berorientasi Objek. Bahasa pemrograman yang digunakan umumnya Java, C#, dan PHP, kemudian mahasiswa akan mempelajari juga bagaimana merancang sebuah sistem atau aplikasi menggunakan UML terhadap suatu kasus dan membuat aplikasinya

ketika memasuki tahap *coding*. Tidak hanya sebatas kuliah, beberapa perguruan tinggi yang mempunyai jurusan informatika, kadang mewajibkan penggunaan UML untuk digunakan sebagai alat bantu di skripsi, tesis, dan disertasi. Karena tidak semua akademisi dapat membaca semua kode program yang datang dari berbagai bahasa pemrograman, maka UML menjadi sarana untuk memahami aplikasi dan sistem yang dikembangkan akademisi lainnya.

Berikut ini adalah beberapa alat yang membantu kita untuk menggunakan UML disebut dengan CASE (*computer aided software engineering*). Ada beberapa CASE yang umum digunakan oleh para profesional yang menggunakan UML, diantaranya:

- Sybase Power Designer
- Rational Rose
- Microsoft Visio
- Enterprise Architect

Ada juga beberapa CASE yang *open source* dan dapat digunakan secara bebas tanpa terbentur dengan masalah pembayaran lisensi seperti berikut:

- StarUML
- DIA Diagram Editor
- Umbrello
- ArgoUML
- Software Ideas Modeller
- Eclipse UML2

Beberapa alat yang berbayar mempunyai kemampuan untuk melakukan *forward engineering* dan *reverse engineering*. *Forward engineering* adalah sebuah fitur dari suatu CASE dimana UML yang telah dirancang dapat menghasilkan kode program

yang akan menjadi pijakan dan panduan untuk mengawali pengembangan aplikasi. Sehingga koherensi antara aplikasi dan UML dapat terjaga. *Reverse engineering* adalah fitur dimana suatu CASE dapat membaca kode program dan menghasilkan UML dari kode program yang telah ada atau sedang dibuat.

2.12 Sistem Pendukung Keputusan

Menurut (Kusrini, 2007) "Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (input) sehingga menghasilkan keluaran (output)."

Menurut Davis dalam Hartono (2013:120) mengemukakan bahwa Terdapat dua model pengambilan keputusan, yaitu model sistem tertutup dan model sistem terbuka.

2.12.1 Model Sistem Tertutup

Model sistem tertutup dilandasi asumsi bahwa keputusan dapat diambil tanpa campur tangan dari lingkungan (luar) sistem, karena sistem pengambilan keputusan tidak dipengaruhi oleh lingkungan. Dalam hal ini sistem pengambilan keputusan dianggap:

1) Mengetahui semua alternatif tindakan untuk menanggapi permasalahan dengan segala konsekuensinya.

2) Memiliki metode untuk menyusun alternatif-alternatif sesuai prioritasnya.

3) Dapat memilih/menetapkan alternatif yang paling menguntungkan, misalnya dari segi laba, manfaat, dan lain-lain.

2.12.2 Model Sistem Terbuka

Model sistem terbuka dilandasi asumsi bahwa sistem pengambilan keputusan dan lingkungan memiliki hubungan saling pengaruh. Keputusan yang diambil akan berdampak terhadap lingkungan dan sebaliknya lingkungan juga

berpengaruh terhadap sistem pengambilan keputusan. Dalam hal ini sistem pengambilan keputusan dianggap:

1) Hanya mengetahui sebagian saja dari alternatif-alternatif untuk menangani permasalahan dengan segala konsekuensinya.

2) Hanya dapat menyajikan sejumlah alternatif yang baik untuk menangani permasalahan, tetapi tidak dapat memilih/menetapkan alternatif yang paling menguntungkan.

3) Sekadar mempersilakan pemilihan alternatif terbaik untuk dilakukan oleh pijak diluar sisten sesuai dengan aspirasinya.

2.13 Metode *Simple Additive Weight* (SAW)

Menurut Fishburn dan MacCrimmon dalam (Munthe, 2013) mengemukakan bahwa Metode *Simple Additive Weight* (SAW), sering juga dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar *metode Simple Additive Weight* (SAW) adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut.

Menurut (Asnawati dan Kanedi, 2012) “Kriteria penilaian dapat ditentukan sendiri sesuai dengan kebutuhan perusahaan.”

$$R_{ij} = \begin{cases} \dfrac{X_{ij}}{\text{Max } X_{ij}} & \text{Jika j adalah atribut keuntungan (benefit)} \\ \dfrac{\text{Min } X_{ij}}{X_{ij}} & \text{Jika j adalah atribut biaya (cost)} \end{cases} \quad \text{..... (1)}$$

Dimana :

Rij = Rating kinerja ternormalisasi

Maxij = Nilai maksimum dari setiap baris dan kolom

Minij = Nilai minimum dari setiap baris dan kolom

Xij = Baris dan kolom dari matriks

Dengan Rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i =1,2,…m dan j = 1,2,…,n.

$$V_i = \sum_{j=1}^{n} W_j R_{ij} \qquad \text{..... (2)}$$

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih. Dimana :

Vi = Nilai akhir dari alternatif

Wi = Bobot yang telah ditentukan

Rij = Normalisasi matriks

Nilai yang lebih besar mengindikasikan bahwa alternatif lebih terpilih. Menurut (Nofriansyah, 2014) Metode *Simple Additive Weighting* (SAW) disarankan untuk menyelesaikan masalah penyeleksian dalam sistem pengambilan keputusan multi proses. Metode *Simple Additive Weight* (SAW) merupakan metode yang banyak digunakan dalam pengambilan keputusan yang memiliki banyak atribut.

Menurut Fishburn dan MacCrimmon dalam (Munthe, 2013) Ada beberapa langkah dalam penyelesaian metode Simple Additive Weight (SAW) adalah sebagai berikut:

1. Menentukan kriteria-kriteria yang dijadikan acuan dalam pendukung keputusan yaitu Ci.

2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.

3. Membuat matriks keputusan berdasarkan kriteria (Ci).

4. Kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan maupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.

5. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai aternatif terbaik (Ai) sebagai solusi.

2.14 Man Month

Dalam kamus bisnis *man month* adalah Waktu kerja satu orang selama satu bulan, atau Unit kerja yang mewakili upaya produktif dari satu orang dalam waktu 4 minggu. Juga disebut bulan kerja.

Di FP5, Uni Eropa menetapkan bahwa terdapat 1.680 jam waktu produktif per tahun atau 140 jam waktu produktif per bulan untuk mereka yang bekerja 5 hari 8 jam per minggu. Di institusi FP7 harus digunakan angka aktual mereka sendiri dan oleh karena itu harus membuat perhitungan sendiri (yang mungkin termasuk jam produktif selain 1.680 jam per tahun). Oleh karena itu terdapat rumus untuk menghitung waktu produktif ialah sebagai berikut

$$PT = (A - LW - NH - l) * WH$$

Dimana:

PT = Productive Time (Waktu Produktif)

A = Annual (Tahunan atau bulanan)

LW = Less Weekends (Hari Libur)

NH = National Holidays (Libur Nasional)

l = leave (cuti / sakit)

WH = Work Hour (waktu kerja dalam sehari)

Berikut merupakan contoh penghitungan *man month*

Man Month Calculation Example

	Days		Hours (based on 8 hour day)
Annual	365		
Less Weekends	104		
Net	261	=	2,088
National/statuary holidays, etc.	- 15		
Annual vacation	- 21		
Annual sick days	- 15		
	51	=	408
Annual net productive time	**210**		**1,680**

Dari contoh penghitungan tersebut dapat ditarik kesimpulan bahwa dalam 1 tahun dengan 104 hari libur kerja , 15 hari libur nasional , 21 hari cuti , dan 15 hari sakit serta perharinya bekerja selama 8 jam. Maka waktu produktif orang tersebut adalah 1680 jam selama 1 tahun.

2.15 Teori Gamifikasi

Gabe Zichermann pada buku *Gamification By Design*, mengatakan bahwa gamifikasi merupakan proses dari cara berpikir seperti *game* dan komponen-komponen yang ada pada *game* untuk melibatkan pengguna dan menyelesaikan masalah tertentu [9]. Komponen-komponen yang ada pada *game* disebut juga *game mechanics.*

2.15.1 *Game Mechanics*

Game mechanics merupakan komponen yang diambil dari konsep *game* dalam penerapan gamifikasi. *Game mechanics* diantaranya adalah :

A. Point

Point merupakan alat tukar yang dimiliki pengguna yang berlaku didalam aplikasi yang di gamifikasi. Point adalah motivator yang efektif yang dapat digunakan sebagai reward/penghargaan ke pengguna, sebagai indikator status, dan sebagai alat

tukar pembelian barang atau jasa yang diatur oleh *project manager* atau organisasi [6]. *Experience point*

Experience point adalah poin yang terus bertambah seiring diselesaikannya suatu aktivitas atau pekerjaan dalam sistem. *Experience point* merupakan poin yang paling penting dari berbagai macam jenis poin. Jenis *point* ini digunakan untuk melihat urutan pengguna. *Experience point* pada umumnya bersifat tidak terbatas, tidak bisa ditukarkan, dan tidak dapat kadaluarsa. Tetapi pada beberapa kasus, *experience point* dapat kadaluarsa dengan tujuan untuk menyamakan kembali kedudukan para pengguna [9].

Experience point biasanya digunakan untuk *role-play game* dimana pada awalnya pengguna tidak berpengalaman. Dengan setiap tugas yang diselesaikan, pengguna mendapat pengalaman berupa *experience point* dan siap untuk menghadapi tugas yang lebih sulit atau rumit [10]. Berikut merupakan contoh didapatkannya *experience point :*



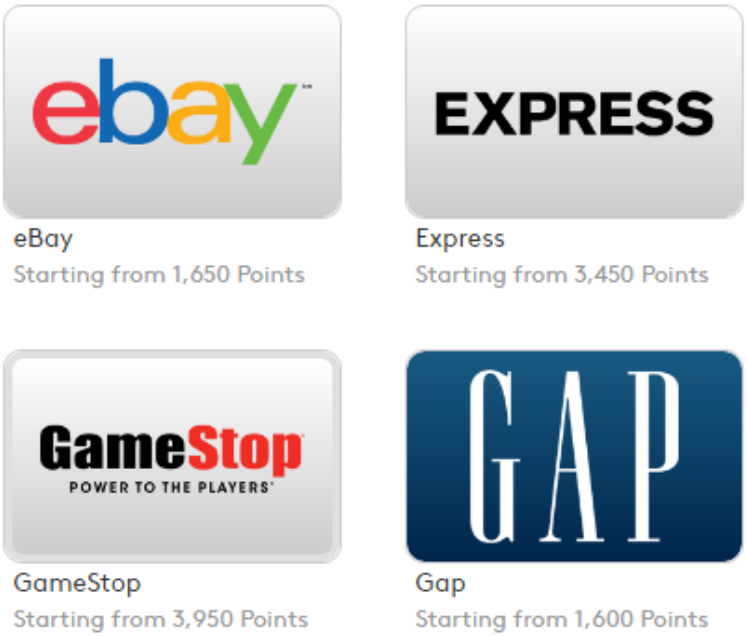
Gambar 2.5 Contoh didapatkannya *experience point*

B. *Redeemable point*

Redeemable point adalah poin yang dapat ditukarkan dengan sesuatu. *Redeemable point* bersifat fluktuatif atau nilainya dapat berkurang atau bertambah. Umumnya membentuk pondasi seperti ekonomi virtual dan sering diberi nama *coins, bucks, cash,* dan lain-lain. Seperti halnya jenis ekonomi lain, *reward point* harus diawasi dan dikelola perputaran nilainya agar memastikan semuanya berjalan dengan baik serta untuk menghindari inflasi atau deflasi [9].

Cara mudah menentukan nilai barang untuk transaksi dalam dollar adalah 1% dari jumlah *redeemable points.* Hal ini telah menjadi standar untuk *loyalty program* dan sudah terbukti efektif [11].

Berikut merupakan contoh penerapan *redeemable point* dengan *reward :*



Gambar 2.6 Contoh penerapan *redeemable point*

C. *Level*

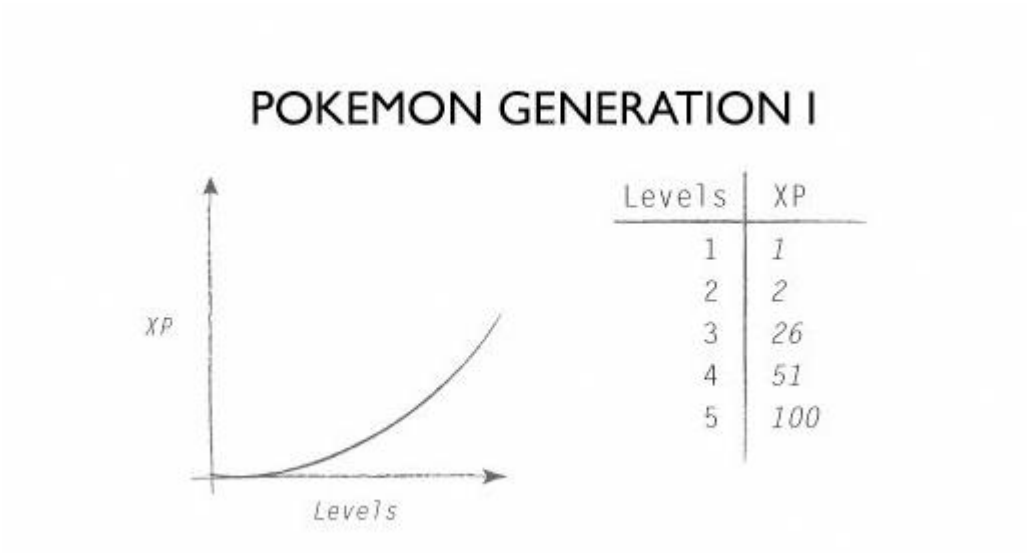
Level adalah suatu angka yang merepresentasikan seluruh pengalaman.. *Level* berfungsi sebagai tanda agar pengguna mengetahui seberapa besar pencapaian dari heasil upaya atau kerja keras yang telah dilakukan. Dalam *game*

design, terdapat karakteristik dari sebuah level yaitu kalkulasi atau penghitungan level selalu memakai fungsi eksponensial. Artinya semakin besar level maka semakin besar juga usaha atau kerja keras yang harus dilakukan.

Berikut merupakan contoh pengimplementasian level dari suatu game.

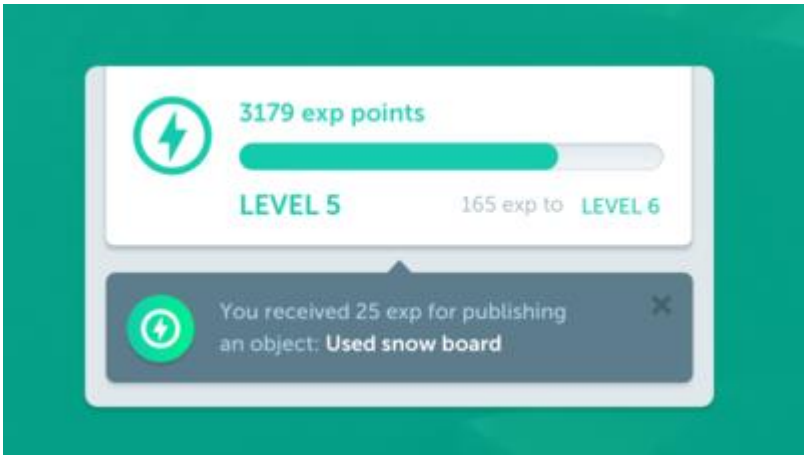
```
function nextLevel(level)
    return round((4 * (level ^ 3)) / 5)
end
```

Gambar 2,7 Contoh fungsi kenaikan level



Gambar 2.8 Contoh implementasi level

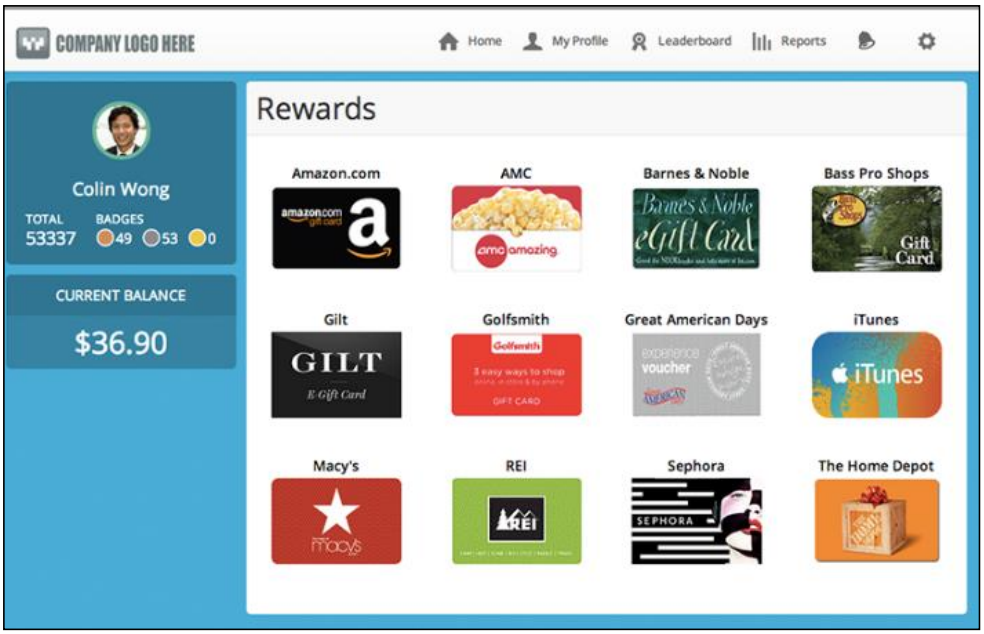
Berikut merupakan contoh penerapan *level* yang berkaitan dengan *experience point* :



Gambar 2.9 Contoh penerapan *level* yang berkaitan dengan *experience point*

D. Reward

Reward merupakan hadiah yang didapat ketika pengguna menukarkan *point* yang dimilikinya. *Reward* tidak berbentuk uang. Agar pengguna tetap ingin menghabiskan *point*-nya atau tetap mengerjakan aktivitas atau pekerjaannya, *reward* harus terus diperbaharui sesuai kebutuhan. Contoh *reward* yang diterapkan pada lingkungan kerja adalah makan malam, voucher belanja, atau voucher liburan [6].

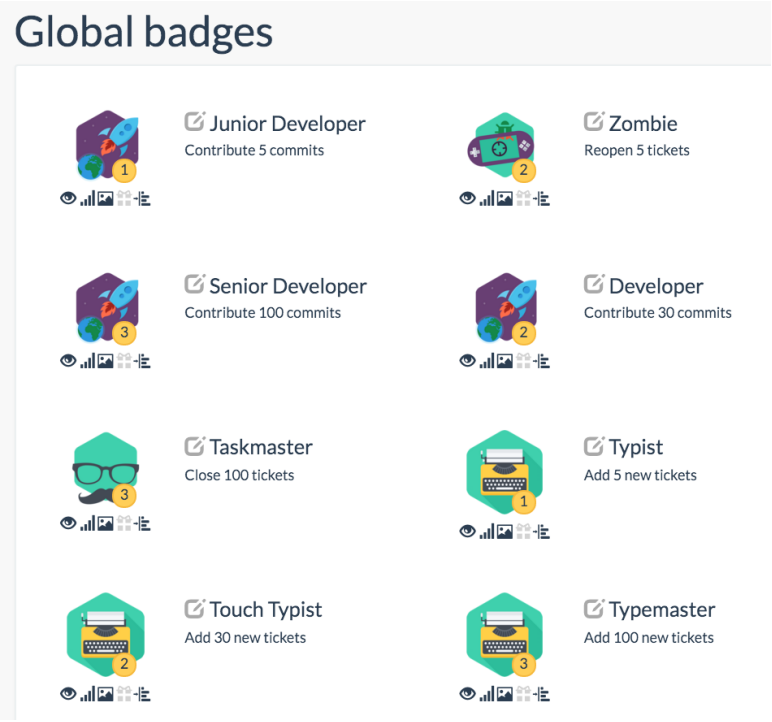


Gambar 2.10 Implementasi sistem *reward* pada *wooboard.com*

E. Badge

Badge ditujukan untuk memperlihatkan kompetensi seorang pengguna sehingga pengguna lain dapat menyadari kemampuan pengguna tersebut. *Badge* berfungsi sama halnya dengan lencana yang diterapkan pada tentara, dimana *badge* mencerminkan pelatihan yang telah dilewati. Pengguna dalam aplikasi yang di-gamifikasi dapat mendapatkan *badge* dan dapat ditunjukan ke pengguna lainnya.

Badge menggambarkan kemampuan seseorang dan melekat pada profil pengguna sebagai rekaman pencapaian yang dapat dilihat semua pengguna. *Badge* digunakan untuk melihat pencapaian pengguna [6].



Gambar 2.11 Contoh *badge* beserta cara mendapatkannya

F. Leaderboard

Pengakuan performa antar pengguna yang berdasarkan dari perolehan *point* dan *badge* merupakan hal yang penting. Salah satu cara untuk memunculkan pengakuan ini adalah melalui fitur *leaderboard. leaderboard* berisi informasi *ranking* pengguna berdasarkan akumulasi *point* yang dimiliki setiap pengguna yang dapat dilihat oleh pengguna lain. Pengguna yang menempati posisi teratas *leaderboard* mendapatkan manfaat berupa pengakuan, status sosial, dan harga diri. Sedangkan bagi pengguna yang berada pada urutan dibawahnya, timbul keinginanan untuk memperbaiki posisinya. Elemen kompetisi ini yang menjadi penggerak orang-orang, untuk memenuhi kebutuhan mereka akan tantangan dan penghargaan, dan memotivasi daya kerja menuju target.

Untuk beberapa anggota dalam tim, elemen dari pengakuan sosial dapat menjadi lebih penting daripada penghargaan fisik. Para individu yang mencari pengembangan karir dapat lebih menguntungkan dengan pengakuan yang diraih melalui gamifikasi. Rekan-rekannya juga dapat melihat progress masing-masing dan menyadari tugas yang telah dikerjakan secara sempurna, sementara manager mempunyai kesempatan untuk melihat performa tim [6]. Berikut merupakan contoh *leaderboard :*



Gambar 2.12 *leaderboard* menampilkan urutan pengguna berdasarkan point

2.15.2 Manfaat Gamifikasi

Berikut merupakan manfaat utama yang dapat ditimbulkan oleh gamifikasi [6] :

a. Meningkatkan keterlibatan atau partisipasi pengguna.
b. Meningkatkan tingkat motivasi pengguna.
c. Meningkatkan interaksi dengan pengguna (klien atau karyawan).
d. Meningkatkan loyalitas pengguna.

2.15.3 Gamifikasi Pada Lingkungan Kerja

Pada lingkungan kerja, motivasi kerja merupakan hal yang harus dimiliki oleh setiap perkerja/pegawai agar mereka dapat memberikan kontribusi positif bagi keberlangsungan perusahaan. Banyak perusahaan yang mengeluarkan banyak uang untuk mengadakan seminar motivasi agar pekerja tidak kehilangan motivasi dalam bekerja [9]. Terdapat beberapa faktor yang menyebabkan pegawai kehilangan motivasi kerja, salah satunya adalah kurangnya pemberian motivasi berupa penghargaan kepada pegawai [9].

Di era komputerisasi ini, banyak *software* yang dibuat untuk membantu proses-proses yang terdapat pada perusahaan sebagai contoh, seperti *software* untuk pengolahan kata/dokumen, penggajian, absensi, hingga software yang bermanfaat untuk membantu pengarsipan dan koordinasi kerja antar pegawai atau tim yaitu *software project management*. Salah satu solusi dalam bentuk aplikasi yang membantu/mengurangi *effort* yang harus dikeluarkan pada proses pemberian motivasi kepada *team member*, adalah dengan menggunakan aplikasi *project management* yang telah di gamifikasi. Saat ini, sudah ada *software Project management* yang mengimplementasikan gamifikasi, yaitu RedCritterTracker. *Game mechanics* yang diterapkan pada aplikas tersebut diantaranya *reward point, lifetime point, badges, rewards,* dan *leaderboard. Game mechanics* tersebut di impelementasikan melalui Fitur-fitur seperti mendapatkan *reward point* dan *lifetime point* saat selesai menyelesaikan suatu tugas, pembelian *reward* menggunakan *reward point,* mendapatkan *badge* saat memenuhi kondisi tertentu,

dan melihat *leaderboard* untuk mengetahui urutan berdasarkan *lifetime point* suatu *user* terhadap *user* lain