

## BAB II

### KAJIAN PUSTAKA

#### 2.1. Penelitian Terdahulu

Dalam mengerjakan penelitian ini, penelitian-penelitian terdahulu yang sudah dilakukan sebelumnya digunakan sebagai kajian serta referensi. Terdapat beberapa penelitian terdahulu dengan topik yang sama yaitu penelitian mengenai penentuan prioritas lokasi pembangunan embung, diantaranya penelitian yang dilakukan oleh Anjasgoro, dkk [2], dengan judul “Analisis Prioritas Pembangunan Embung Metode *Cluster Analysis*, AHP, dan *Weighted Average* (Studi Kasus: Embung di Kabupaten Semarang)”, penelitian tersebut menggunakan 3 metode yang berbeda untuk menganalisis penentuan embung prioritas, yaitu metode *Cluster Analysis*, metode AHP, serta *Weighted Average*. Kesimpulan dari penelitian tersebut adalah didapatkan sejumlah variabel yang memiliki pengaruh dalam pembangunan embung yakni vegetasi area genangan embung, volume material timbunan, volume tampungan efektif, luas daerah yang akan dibebaskan, lama operasi, biaya air/m<sup>3</sup>, akses jalan masuk menuju site bendungan, biaya konstruksi embung, biaya operasional, cakupan daerah irigasi, status lahan di *site* dan genangan, dan manfaat air baku [2].

Penelitian terdahulu lainnya mengenai embung adalah penelitian yang berjudul “*Analysis of Fuzzy TOPSIS Method in Determining Priority of Small Dams Construction*” yang dilakukan oleh Desyta Ulfiana dan Suharyanto [10]. Penelitian tersebut dilakukan untuk mengetahui prioritas pembangunan bendungan kecil di Kabupaten Semarang menggunakan metode TOPSIS. Aspek teknis atau kriteria yang digunakan yaitu vegetasi area genangan embung, volume material timbunan, volume tampungan efektif, luas daerah yang akan dibebaskan, lama operasi, biaya air/m<sup>3</sup> dan akses jalan menuju situs bendungan. Untuk mengakomodasi jenis kriteria yang memiliki variabel linguistik, logika *fuzzy* digunakan untuk mengukur. Logika *fuzzy* kemudian diimplementasikan dalam metode TOPSIS sehingga analisis terbaik dapat diperoleh [10].

Penelitian terdahulu mengenai metode VIKOR antara lain penelitian dari Salvius Paulus Lengkong dkk [7] yang berjudul “Implementasi Metode VIKOR untuk Seleksi Penerima Beasiswa”. Dalam penelitian tersebut, metode VIKOR digunakan untuk mendukung proses seleksi penerimaan beasiswa kepada 40 siswa berdasarkan beberapa kriteria yang telah ditentukan, seperti indeks prestasi (IP), semester, daya listrik, dan tagihan listrik rumah [7].

Penelitian lain mengenai VIKOR juga pernah dilakukan oleh Miftahul Arif [8] dengan judul “*Multi-Criteria Decision Making with the VIKOR and SMARTER Methods for Optimal Seller Selection from Several E-Marketplaces*” yang menggunakan metode VIKOR untuk memilih penjual yang optimal dari beberapa *marketplace*. Selain menggunakan metode VIKOR, dalam penelitian ini juga menggunakan metode SMARTER yang digunakan untuk menentukan level prioritas tiap kriteria yang dibutuhkan untuk melakukan perhitungan ROC (*Rank Order Centroid*) yang mana dengan melakukan kuesioner kepada responden yang berkompeten [8].

Penelitian lainnya mengenai VIKOR adalah penelitian yang berjudul “*Multi-Criteria Optimization of Insulation Options for Warmth of Buildings to Increase Energy Efficiency*” [9]. Penelitian tersebut bertujuan untuk memilih material yang terbaik untuk digunakan sebagai material insulasi pada bangunan. Opsi alternatif bahan insulasi yang dipertimbangkan yaitu *styrofoam*, *mineral wool*, *pluto panels*, *polyester*, *polyurethane*, *perlite*, dan *wood wool* dengan kriteria yang ditentukan adalah harga bahan insulasi, emisi, koefisien konduktivitas termal, kalor spesifik, faktor ketahanan difusi uap air, dan kepadatan. Dalam penelitian ini metode VIKOR digunakan untuk mendapatkan material insulasi terbaik guna memaksimalkan efisiensi energi yang digunakan dan mengurangi biaya dan emisi CO<sub>2</sub>. Dari penelitian ini didapatkan bahwa material *styrofoam* menjadi alternatif terbaik untuk digunakan sebagai material insulasi pada bangunan [9].

Metode VIKOR atau *Višekriterijumsko Kompromisno Rangiranje* yang diimplementasikan ke dalam sebuah sistem pendukung keputusan diharapkan dapat digunakan untuk membantu menentukan solusi dari penentuan prioritas pembangunan embung di Kabupaten Semarang dengan efektif dan efisien.

## 2.2. Embung

Embung merupakan salah satu teknologi konservasi air yang dibangun sebagai solusi/pemecah kekeringan dengan menampung air hujan serta air limpasan (*run off*) sebagai cadangan persediaan pada musim kemarau. Embung merupakan waduk berukuran mikro yang dibangun untuk digunakan sebagai pengendali kelebihan air ketika musim penghujan dengan teknik pemanenan air (*water harvesting*) dan menjadi sumber air untuk irigasi ketika kemarau. Embung juga memiliki tugas untuk mengumpulkan air dari Daerah Pengaliran Sungai (DPS) di bagian hulu yang berasal dari limpasan air hujan [6].

## 2.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) ialah serangkaian kelas tertentu dari sistem informasi terkomputerisasi yang mendukung kegiatan pengambilan keputusan. Aplikasi Sistem Pendukung Keputusan (SPK) mengimplementasikan sistem informasi berbasis komputer atau *Computer Based Information Systems* (CBIS) untuk membantu menyelesaikan masalah melalui penyediaan solusi yang efektif [11].

Sistem Pendukung Keputusan (SPK) merupakan sebuah sistem yang mampu memberikan kemampuan pemecahan masalah, sistem ini dibuat untuk meningkatkan produktivitas dengan melakukan proses pembuatan keputusan secara otomatis [12].

Karakteristik dari sistem pendukung keputusan [11] antara lain :

- a. Membantu dalam proses pengambilan keputusan bagi perusahaan atau organisasi.
- b. Terdapat antarmuka agar mudah dipahami bagi pengguna dengan pengguna tetap memegang kendali terhadap proses pengambilan keputusan.
- c. Membantu dalam pengambilan keputusan pada masalah terstruktur dan semi terstruktur juga mendukung sejumlah keputusan yang saling terkait dan berinteraksi satu sama lain.
- d. Memiliki kemampuan dialog untuk mengumpulkan informasi yang sesuai dengan kebutuhan pengguna.

- e. Terdapat sub-sistem yang saling terintegrasi dengan baik sehingga mampu bekerja sebagai satu kesatuan sistem yang utuh.
- f. Terdapat dua buah komponen utama yakni data dan model.

Tujuan dari sistem pendukung keputusan [11] adalah:

1. Membantu memberikan solusi dalam pengambilan keputusan pada suatu masalah.
2. Menyediakan dukungan atas pertimbangan yang dilakukan oleh manajer tanpa bertujuan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan diambil daripada perbaikan efisiensinya.
4. Kecepatan proses perhitungan oleh komputer memungkinkan pengambil keputusan melakukan banyak perhitungan secara cepat dengan biaya yang ekonomis.
5. Peningkatan produktivitas dengan membangun suatu kelompok pengambil keputusan terutama pakar bisa sangat mahal. Namun dengan sistem pendukung keputusan komputerisasi, ukuran kelompok bisa dikurangi dan anggotanya dapat berada di lokasi yang berbeda-beda, sehingga memperkecil biaya perjalanan.

#### **2.4. *Višekriterijumsko Kompromisno Rangiranje* (VIKOR)**

*Višekriterijumsko Kompromisno Rangiranje* atau VIKOR dalam bahasa Serbia yang berarti “perangkingan kompromis multi-kriteria” merupakan salah satu metode pengambilan keputusan multikriteria. Landasan dari solusi kompromi dalam VIKOR dibuat oleh Yu (1973) dan Zeleny (1982) kemudian diteruskan oleh Opricovic dan Tzeng (2002, 2003, 2004, dan 2007) [13]. Metode VIKOR termasuk dalam kategori metode *Multi-Criteria Decision Making* (MCDM) yang telah dipakai secara luas guna menyelesaikan berbagai macam pengambilan keputusan berdasarkan banyak kriteria dengan mengajukan solusi kompromi berdasarkan solusi ideal yang diperkirakan. Metode VIKOR memiliki kemampuan untuk mengatasi kriteria yang saling bertentangan saat melakukan proses perangkingan, maksud kriteria bertentangan disini adalah tiap kriteria dapat memakai penilaian

berbeda dengan kriteria yang lain yakni kriteria dapat menggunakan tren *benefit* (semakin tinggi nilainya maka semakin baik) atau tren *cost* (semakin kecil nilainya maka semakin baik). Metode VIKOR sendiri memiliki kelemahan dalam melakukan pembobotan kriteria karena tidak ada perhitungan khusus untuk menghitung nilai bobot setiap kriteria, pembobotan kriteria dalam metode VIKOR diberikan begitu saja oleh pengambil keputusan sehingga diperlukan metode lain untuk memeriksa konsistensi bobot [8].

Terdapat beberapa tahap yang dilakukan dalam penyelesaian masalah menggunakan metode VIKOR [7][8]. Tahap-tahap tersebut meliputi:

#### 1. Menyusun Matriks Keputusan ( $F$ )

Setiap alternatif dan kriteria disusun ke dalam bentuk matriks keputusan ( $F$ ).  $A_i$  menyatakan alternatif ke 1,2,3, ...,  $i$  dan  $C_{xj}$  menyatakan kriteria ke 1,2,3, ...,  $j$  dan  $x_{ij}$  menyatakan respons alternatif  $i$  pada kriteria  $j$ . Matriks keputusan ( $F$ ) ditunjukkan pada persamaan 2.1 berikut:

$$F = \begin{matrix} & \begin{matrix} C_{x1} & C_{x2} & \cdots & C_{xj} \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_i \end{matrix} & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1j} \\ x_{21} & x_{22} & \cdots & x_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} \end{pmatrix} \end{matrix} \quad (2.1)$$

Keterangan:

$F$  = Matriks keputusan

$A_i$  = Alternatif ke -  $i$

$C_j$  = Kriteria ke -  $j$

$x_{ij}$  = Respons alternatif  $i$  pada kriteria  $j$

$i$  = 1,2,3, ...,  $i$  adalah nomor urutan alternatif

$j$  = 1,2,3, ...,  $j$  adalah nomor urutan kriteria

#### 2. Menentukan Bobot Kriteria ( $W$ )

Menentukan bobot kriteria yang diperoleh dari pengguna sistem sesuai dengan kebutuhan atau kriteria yang diinginkan. Rumusan umum untuk bobot kriteria ditunjukkan pada persamaan 2.2 berikut:

$$\sum_{j=1}^n W_j = 1 \quad (2.2)$$

Keterangan:

$W_j$  = Bobot kriteria  $j$

### 3. Membuat Matriks Normalisasi ( $N$ )

Membuat matriks normalisasi dengan menentukan nilai positif ( $f_j^+$ ) dan nilai negatif ( $f_j^-$ ) sebagai solusi ideal untuk setiap kriteria. Penentuan nilai data terbaik/positif ( $f_j^+$ ) dan terburuk/negatif ( $f_j^-$ ) atau dengan istilah *Cost* dan *Benefit* ditentukan oleh jenis data kriteria apakah *higher-the-better* (HB) atau *lower-the-better* (LB). Untuk menentukan nilai positif dan nilai negatif masing-masing kriteria digunakan ketentuan sebagai berikut:

- Jika kriteria memiliki tren *benefit*, maka gunakan persamaan 2.3 dan persamaan 2.4 berikut:

$$f_j^+ = \max(f_{1j}, f_{2j}, f_{3j}, \dots, f_{ij}) \quad (2.3)$$

$$f_j^- = \min(f_{1j}, f_{2j}, f_{3j}, \dots, f_{ij}) \quad (2.4)$$

- Jika kriteria memiliki tren *cost*, maka gunakan persamaan 2.5 dan persamaan 2.6 berikut:

$$f_j^+ = \min(f_{1j}, f_{2j}, f_{3j}, \dots, f_{ij}) \quad (2.5)$$

$$f_j^- = \max(f_{1j}, f_{2j}, f_{3j}, \dots, f_{ij}) \quad (2.6)$$

Selanjutnya melakukan normalisasi pada matriks  $F$  dengan persamaan 2.7 berikut:

$$N_{ij} = \frac{(f_j^+ - f_{ij})}{(f_j^+ - f_j^-)} \quad (2.7)$$

Keterangan:

$N$  = Matriks ternormalisasi

$f_{ij}$  = Fungsi respons alternatif  $i$  pada kriteria  $j$

$f_j^+$  = Nilai terbaik dalam satu kriteria  $j$

$f_j^-$  = Nilai terjelek dalam satu kriteria  $j$

#### 4. Menghitung Normalisasi Bobot ( $F^*$ )

Menghitung nilai terbobot dari data ternormalisasi untuk setiap alternatif terhadap kriteria dengan melakukan perkalian antara nilai data ternormalisasi ( $N_{ij}$ ) dengan nilai bobot kriteria ( $W_j$ ) yang telah ditentukan, persamaan perhitungan normalisasi bobot ditunjukkan pada persamaaan 2.8 sebagai berikut:

$$F_{ij}^* = W_j \cdot N_{ij} \quad (2.8)$$

Keterangan:

$F_{ij}^*$  = Nilai data ternormalisasi sudah terbobot untuk alternatif  $i$  pada kriteria  $j$

$W_j$  = Nilai bobot kriteria  $j$

$N_{ij}$  = Nilai data ternormalisasi untuk alternatif  $i$  pada kriteria  $j$

#### 5. Menghitung Nilai *Utility Measure* ( $S$ ) dan *Regret Measure* ( $R$ )

Menghitung nilai *utility measure* ( $S$ ) dan *regret measure* ( $R$ ) untuk setiap alternatif yang mana nilai  $S_i$  menyatakan nilai jarak alternatif ke solusi ideal positif sedangkan  $R_i$  menyatakan nilai jarak alternatif ke solusi ideal negatif. Untuk menghitung  $S_i$  dan  $R_i$  digunakan persamaan 2.9 dan persamaan 2.10 sebagai berikut:

$$S_i = \sum_{j=1}^n F_{ij}^* \quad (2.9)$$

$$R_i = \max_j [F_{ij}^*] \quad (2.10)$$

Keterangan:

$S_i$  = Nilai *Utility Measure* untuk alternatif ke -  $i$

$R_i$  = Nilai *Regret Measure* untuk alternatif ke -  $i$

$F_{ij}^*$  = Nilai data ternormalisasi sudah terbobot untuk alternatif  $i$  pada kriteria  $j$

#### 6. Menghitung Nilai Indeks VIKOR ( $Q$ )

Melakukan perhitungan nilai indeks VIKOR ( $Q$ ) untuk setiap alternatif dengan menggunakan nilai  $S_i$ ,  $S^+$ ,  $S^-$ ,  $R_i$ ,  $R^+$ , dan  $R^-$  yang didapat dari perhitungan

*utility measures* dan *regret measure* serta nilai  $V$  yang merupakan bobot yang nilainya antara 0 hingga 1 (umumnya bernilai 0,5). Nilai  $V$  merupakan bobot *strategy of the maximum group* sementara nilai  $(1 - V)$  adalah bobot *individual regret*. Semakin rendah nilai  $Q_i$ , maka semakin baik solusi alternatif tersebut. Untuk mencari nilai  $Q_i$  digunakan persamaan 2.11 sebagai berikut:

$$Q_i = V \left[ \frac{(S_i - S^-)}{(S^+ - S^-)} \right] + (1 - V) \left[ \frac{(R_i - R^-)}{(R^+ - R^-)} \right] \quad (2.11)$$

Keterangan:

$Q_i$  = Nilai Indeks VIKOR alternatif

$V$  = Bobot berkisar antara 0-1 (umumnya bernilai 0.5)

$S^+$  =  $\max_i(S_i)$

$S^-$  =  $\min_i(S_i)$

$R^+$  =  $\max_i(R_i)$

$R^-$  =  $\min_i(R_i)$

## 7. Perangkingan Alternatif

Setelah menghitung nilai  $Q_i$ , maka terdapat 3 perangkingan:  $S_i$ ,  $R_i$ , dan  $Q_i$ . Solusi kompromi dilihat pada perangkingan  $Q_i$ . Perangkingan ditentukan dari nilai  $Q_i$  yang paling rendah sebagai solusi ideal.

## 8. Mengajukan Solusi Kompromi

Solusi kompromi ditentukan dari alternatif yang memiliki peringkat terbaik dengan mengukur indeks VIKOR yang minimum dengan mengujinya dengan 2 kondisi berikut:

- Kondisi 1: *Acceptable Advantage*

Menghitung selisih antara peringkat alternatif pertama dan kedua yakni  $Q_{(a_1)}$  dan  $Q_{(a_2)}$  lalu membandingkannya dengan nilai  $DQ$ . Jika nilai selisih yang didapat lebih besar atau sama dengan nilai  $DQ$ , maka kondisi *acceptable advantage* terpenuhi. Persamaan dari kondisi *acceptable advantage* ditunjukkan pada persamaan 2.12 dan persamaan 2.13 sebagai berikut:

$$Q_{(a_2)} - Q_{(a_1)} \geq DQ \quad (2.12)$$



$$DQ = \frac{1}{m - 1} \quad (2.13)$$

Keterangan:

$Q_{(a_2)}$  = Alternatif peringkat kedua

$Q_{(a_1)}$  = Alternatif peringkat pertama

$m$  = jumlah alternatif

- Kondisi 2: *Acceptable Stability in Decision Making*

Pengujian kondisi 2 dilakukan dengan menguji stabilitas perangkingan alternatif dengan menggunakan 3 nilai  $V$  yang berbeda-beda yakni: nilai  $V > 0,5$  (*voting by majority rule*), nilai  $V = 0,5$  (*by concensus*), dan nilai  $V < 0,5$  (*with veto*) pada perhitungan nilai indeks VIKOR (persamaan 2.11). Umumnya nilai  $V_1$ , nilai  $V_2$ , dan nilai  $V_3$  berkisar antara 0 sampai 1, sedangkan penentuan nilai  $V_1$  dan nilai  $V_3$  untuk pengujian kondisi 2 umumnya berlaku persamaan yang ditunjukkan pada Tabel 2.1 sebagai berikut:

Tabel 2.1 Persamaan penentuan tiga nilai  $V$

Nilai $V_1$	Nilai $V_2$	Nilai $V_3$
$0,5 - x$	$0,5$	$0,5 + x$

Penentuan nilai  $V$  sangat berpengaruh terhadap hasil dari pengujian kondisi 2, pemberian nilai  $V$  yang tidak sesuai dengan persamaan pada Tabel 2.1 dapat menghasilkan pengujian kondisi 2 yang kurang akurat. Selanjutnya, setelah dilakukan perhitungan dengan 3 nilai  $V$  yang berbeda, jika alternatif peringkat pertama atau  $Q_{(a_1)}$  tetap menjadi peringkat terbaik dalam 3 macam pemeringkatan dengan nilai  $V$  yang berbeda, maka kondisi *acceptable stability in decision making* terpenuhi.

Apabila salah satu kondisi tidak terpenuhi, maka solusi kompromi dapat diajukan sebagai berikut:

- Apabila hanya kondisi 2 yang tidak terpenuhi, maka memilih alternatif peringkat pertama dan kedua atau  $Q_{(a_1)}$  dan  $Q_{(a_2)}$ .

- Apabila kondisi 1 tidak terpenuhi, maka memilih alternatif  $Q_{(a_1)}, Q_{(a_2)}, \dots, Q_{(a_m)}$ . Dimana alternatif  $Q_{(a_m)}$  ditentukan dengan persamaan 2.14 dan persamaan 2.13 sebagai berikut:

$$Q_{(a_m)} - Q_{(a_1)} < DQ \quad (2.14)$$

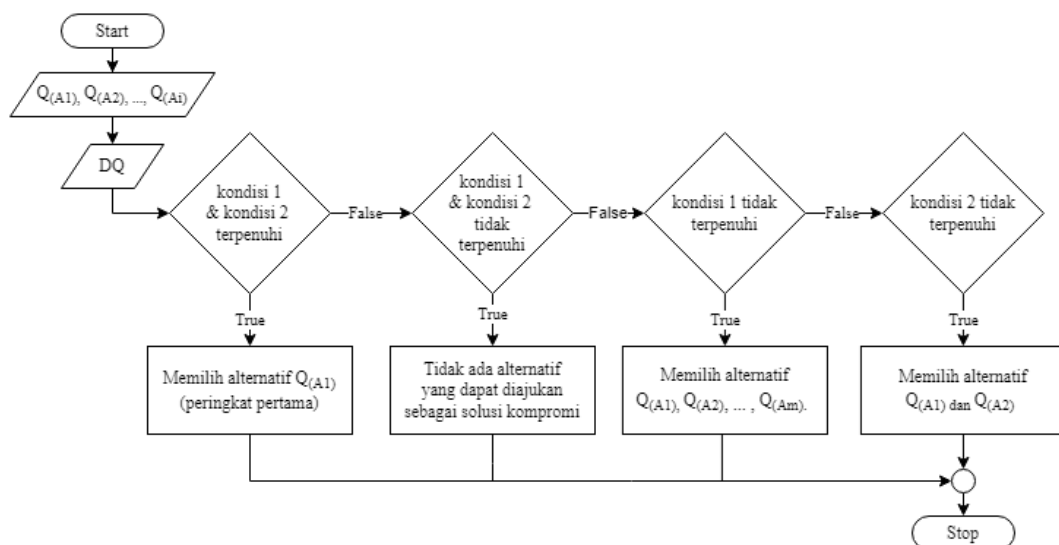
$$DQ = \frac{1}{m - 1} \quad (2.13)$$

Keterangan:

$Q_{(a_m)}$  = Alternatif  $m$  maksimum yang berada dalam kondisi berdekatan

$m$  = jumlah alternatif

Diagram alur untuk pengajuan solusi kompromi ditunjukkan pada Gambar 2.1 sebagai berikut:



Gambar 2.1 Diagram alur pengajuan solusi kompromi metode VIKOR

## 2.5. Basis Data

Basis data terdiri dari dua kata yakni basis dan data. Basis bisa diartikan sebagai suatu markas, tempat bersarang atau tempat berkumpul. Sedangkan data merupakan representasi dari fakta dunia yang mewakili suatu objek (barang, manusia, keadaan, peristiwa, dan sebagainya) yang direkam dalam bentuk huruf, angka, teks, simbol, bunyi, gambar atau kombinasinya. Bisa disimpulkan basis data

adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah [14].

Basis data merupakan salah satu komponen yang sangat penting dalam sebuah sistem informasi. Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Basis data memiliki tujuan untuk mengatur data sehingga didapatkan kecepatan dan kemudahan (*speed*), ketersediaan (*availability*), keakuratan (*accuracy*), kelengkapan (*completeness*), keamanan (*security*), efisiensi ruang penyimpanan (*space*), dan kebersamaan (*shareability*) [14].

## 2.6. CodeIgniter

Codeigniter ialah sebuah aplikasi *open source* yang berupa *framework* atau kerangka kerja yang digunakan untuk membangun *website* dengan bahasa pemrograman PHP. Tujuan penggunaan *framework* adalah untuk mempercepat pengembangan proyek dengan menyediakan *library* yang dapat digunakan dibanding penulisan kode dasar atau kode terstruktur. Codeigniter relatif mudah digunakan dan dipelajari karena memiliki *interface* yang sederhana, dokumentasi yang lengkap, dan *library* atau kumpulan *tools* yang dapat digunakan untuk membuat sebuah proses kerja untuk *website* yang dibuat [15].

Codeigniter menggunakan pola desain MVC (*Model, View, Controller*). MVC merupakan suatu konsep yang memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi, yakni [16]:

1. *Model*: merupakan bagian yang berhubungan secara langsung dengan basis data untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari *controller*, namun tidak berhubungan langsung dengan *view*.
2. *View*: bagian yang menangani *presentation logic*, pada suatu aplikasi web biasanya berupa file HTML. *View* berfungsi untuk menerima dan merepresentasikan data kepada pengguna. Bagian ini tidak memiliki akses langsung terhadap *model*.

3. *Controller*: bagian yang mengatur hubungan antara bagian *model* dan bagian *view*. Controller berfungsi untuk menerima permintaan (*request*) dan data dari pengguna.

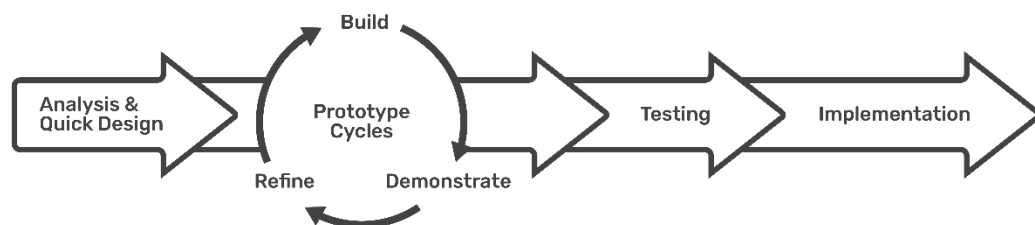
Beberapa keuntungan yang bisa didapatkan dari penggunaan Codeigniter antara lain sebagai berikut [17]:

- Codeigniter berukuran kecil, cepat, sederhana, dan mudah dipelajari.
- Mudah saat melakukan migrasi dari satu server ke server yang lain, cukup hanya mengubah *URL*.
- Proses instalasi yang mudah.
- Dokumentasi yang lengkap dan dukungan komunitas yang sudah kuat.
- Koleksi pustaka atau *library* yang tersedia sudah banyak.

## 2.7. *Rapid Application Development*

*Rapid Application Development* (RAD) adalah sebuah model proses pengembangan aplikasi atau sistem informasi yang mengadopsi pendekatan cepat yang merupakan adaptasi dari model *waterfall*. Metode ini menggunakan pendekatan konstruksi komponen dalam pengembangan aplikasi. Dengan menggunakan metode RAD, pengembangan aplikasi dapat dilakukan dalam waktu yang lebih singkat [18].

Langkah-langkah dalam pengembangan aplikasi menggunakan metode *Rapid Application Development* ditunjukkan pada Gambar 2.2.



Gambar 2.2 *Rapid Application Development*

- a. *Analysis and Quick Design*: mengidentifikasi kebutuhan aplikasi dan masalah untuk menentukan tujuan, batasan-batasan sistem, kendala dan juga alternatif pemecahan masalah.

- b. *Build*: membangun aplikasi sesuai dengan kebutuhan yang sudah diidentifikasi dari pengguna.
- c. *Demonstrate*: melakukan demo kepada *user* untuk mencoba sistem aplikasi.
- d. *Refine*: melakukan perbaikan pada sistem aplikasi jika ada kekurangan yang ditemukan pada langkah *demonstrate*.
- e. *Testing*: menguji sistem aplikasi meliputi fitur, fungsi, *interface*, sampai keseluruhan aspek dari sistem.
- f. *Implementation*: proses implementasi dan finalisasi sistem aplikasi menjadi sebuah produk.