BAB 2
TINJAUAN PUSTAKA

2.1. Tinjauan Instansi

Tinjauan instansi akan mengetengahkan tentang sejarah, visi dan misi, dan struktur organisasi yang ada pada MAN Binong Kabupaten Subang.

2.1.1. Sejarah MAN Binong Kabupaten Subang

MAN Binong dalam prosesnya merupakan penegerian dari MAS Darul Ma'arif Pamanukan berdasarkan Surat Keputusan Menteri Agama No. 515A Tanggal 25 November 1995 dengan nama MAN Pamanukan dengan lokasi di lingkungan Yayasan Darul Ma'arif Pamanukan. Namun dalam perjalanannya, staf guru Yayasan Pendidikan Islam Darul Ma'arif Pamanukan menolak keberadaan MAN Pamanukan, sehingga sedikit atau banyaknya mengakibatkan proses KBM terganggu. Oleh karena itu Drs. Icang Sudaryat (Kepala MAN pada waktu itu) berusaha mencari tempat lain sampai akhirnya bertemu dengan Bpk. Abdurrahman (Humas MTs Darul Hikam Binong). Dari Bpk. Abdurrahman inilah diperoleh petunjuk bahwa ada salah seorang yang hendak mewakafkan tanahnya untuk dijadikan lokasi pendidikan. Hamba Allah yang baik dan hendak mewakafkan tanahnya tersebut tersebut bernama H. Sonjaya. Tanah wakaf tersebut seluas 10.000 M^2 (sepuluh ribu meter persegi). Sesuai akta ikrar wakaf tertanggal 28 November 1995 No. W2/07/157/1995 yang dibuat dihadapan M.I. Haris (Kepala KUA Kec. Binong) dan surat pengesahan Nadzir maka tanah tersebut digunakan untuk keperluan pembangunan Madrasah Aliyah Negeri (MAN) Pamanukan. Sebelum MAN Pamanukan didirikan di atas tanah wakaf

tersebut, maka proses KBM sempat berpindah-berpindah tempat di KUA Kec. Binong, MTs Darul Hikam dan MTs Darul Fikri Ulum.

Pada Tahun 1998 dibangunlah gedung sekolah yang cukup representatif di tanah wakaf tersebut sebanyak 6 ruang belajar yang berlokasi di Desa Cicadas Kecamatan Binong Kabupaten Subang dengan tetap memakai nama MAN Pamanukan. Mengingat nama MAN Pamanukan dirasa janggal, karena lokasi MAN Pamanukan berada di wilayah Binong maka dikeluarkanlah Peraturan Menteri Agama Republik Indonesia No. 4 Tahun 2006 tentang Perubahan Atas lampiran Keputusan Menteri Agama No. 515A Tahun 1995 tentang Pembukaan dan Penegerian beberapa Madrasah yang mengubah nama MAN Pamanukan menjadi MAN Binong. Sesuai dinamika waktu yang telah mencapai selama 13 tahun, MAN Binong mengalami perkembangan baik secara fisik dan non fisik dan berusaha untuk berbenah sesuai dengan tuntutan dinamika pendidikan yang senantiasa berubah. Menyadari akan hal tersebut, maka MAN Binong berusaha untuk melakukan inovasi sesuai dengan perubahan zaman guna mempersiapkan lulusannya untuk menempuh pendidikan yang lebih tinggi dan sekaligus membekali untuk hidup bermasyarakat.

2.1.2. Visi dan Misi

Kurikulum tingkat satuan pendidikan Madrasah Aliyah Negeri Binong Kabupaten Subang disusun dalam kerangka manifestasi visi dan misi yang menjadi cita-cita agung institusi. Adapun visi dan misi lembaga dirumuskan sebagai berikut :

a. Visi

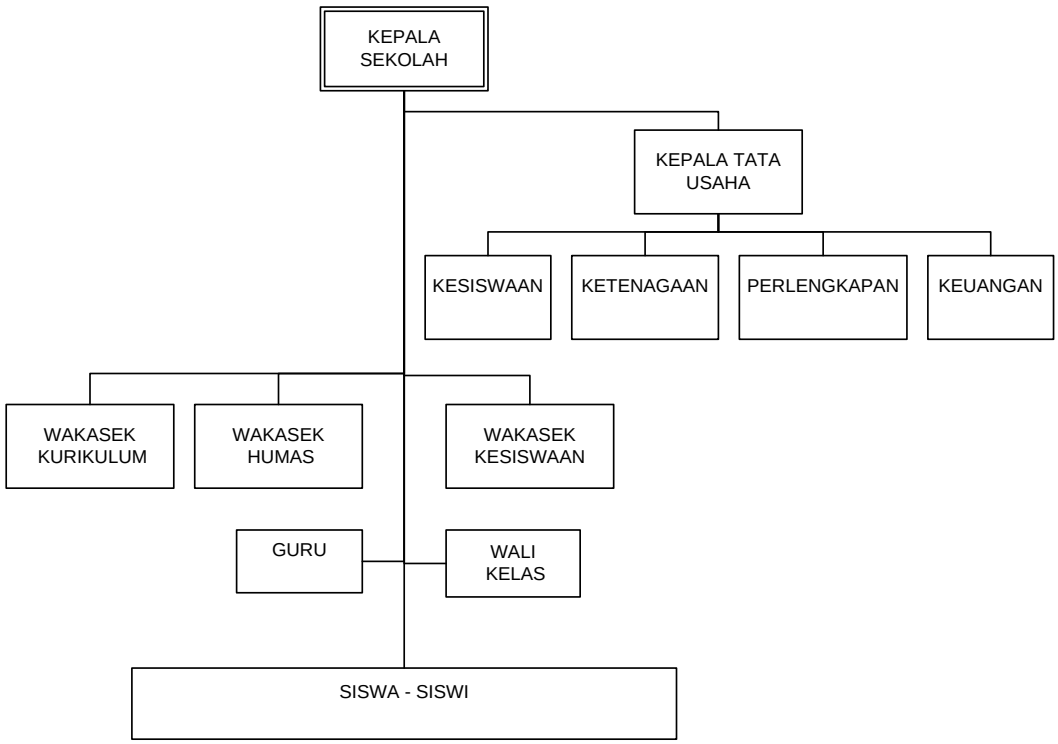
Terdidik dalam nuansa Islami, unggul dalam prestasi dan berakhlakul karimah.

b. Misi

1. Pemantapan sikap dan kepribadian yang agamis dalam kehidupan sehari-hari melalui pembiasaan pengamalan keagamaan.
2. Melaksanakan pembelajaran yang berbasis kemampuan sehingga siswa berkembang sesuai dengan bakat, minat dan kemampuannya.
3. Membantu setiap siswa memahami tentang kemampuan dan potensi diri serta pengembangannya secara optimal.
4. Menumbuhkembangkan semangat kompetitif yang sehat dalam meraih keunggulan prestasi pada semua warga madrasah.
5. Mengoptimalkan pemanfaatan lahan/tanah dan pengadaan fasilitas pendukung pembelajaran.
6. Pemahaman tentang pemanfaatan hasil teknologi (komputer, internet) bagi pengembangan ilmu pengetahuan.
7. Menerapkan manajemen berbasis madrasah yang melibatkan semua *stakeholder*.

2.1.3. Struktur Organisasi

Struktur Organisasi MAN Binong Kabupaten Subang sebagaimana tertuang dalam Surat Keputusan Direksi Nomor: KEP/190/082007 bulan Agustus 2006 tentang Struktur Organisasi dan Tata MAN Binong Kabupaten Subang, adalah seperti pada gambar 2.1



Gambar 2.1 Stuktur Organisasi MAN Binong Subang

Uraian Struktur Organisasi

Uraian tentang struktur organisasi yang berada di bawahnya baik struktural maupun fungsional adalah sebagai berikut :

1. Kepala Sekolah dalam memimpin dan mengelola sekolah mempunyai fungsi perencanaan, pengembangan, dan penetapan kebijakan umum sekolah berdasarkan prinsip kehati-hatian, efektif dan efisen, sesuai dengan visi, misi dan tujuan sekolah. Mengoordinasikan kegiatan para guru serta pengembangan kebijakan komunikasi, pengawasan intern, ksetramatan sekolah dan peningkatan kesejahteraan siswa/i.
2. Kepala Tata Usaha mempunyai fungsi membuat kajian dan memberikan usulan kepada Kepala sekolah mengenai hal-hal yang berkaitan dengan bidang kesiswaan, ketenagaan, perlengkapan dan keuangan.

3. Wakasek Kurikulum mempunyai tanggung jawab tentang silabus dan kurikulum setiap semester dan perkelasnya.
4. Wakasek Humas mempunyai tanggung jawab tentang hubungan baik itu interen maupun eksteren.
5. Wakasek Kesiswaan mempunyai tanggung jawab tentang kegiatan siswa/i baik dalam bidang formal maupun nonformal.
6. Wali kelas adalah orang yang bertanggung jawab terhadap anak kelasnya.
7. Guru adalah orang bertanggung jawab memberikan materi kepada siswa.

2.2. Landasan Teori

Landasan teori merupakan ilmu pendukung yang dapat dijadikan referensi atau pegangan dalam pembangunan sistem ini.

2.2.1 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut ini: [7]

> *Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.*

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut ini: [7]

> *Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.*

2.2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik yang tertentu (gambar 2.2), yaitu : [7]

1. Komponen Sistem (*Components*)

   Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian sistem, yang mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem keseluruhan.

2. Batas Sistem (*Boundary*)

   Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

   Lingkungan luar (*environments*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan merugikan sistem.

4. Penghubung Sistem (*Interface*)

   Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan susbsistem lainnya sehingga memungkinkan sumber-sumber daya mengalir antara subsistem yang satu dengan yang lain.

5. Masukan Sistem (*Input*)

   Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

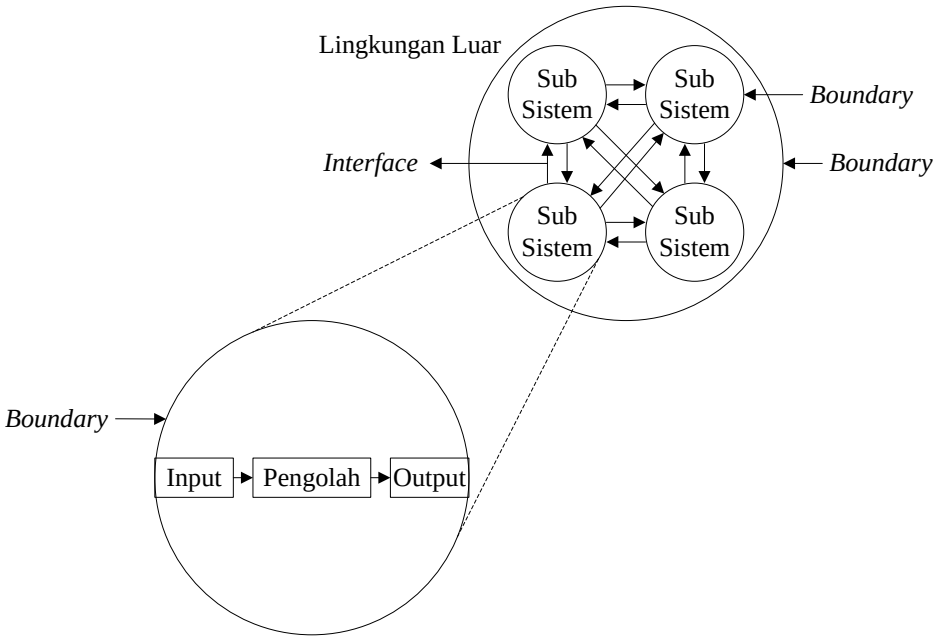
   Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan jadi keluaran.

8. Sasaran Sistem (*Objectives*)

   Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar 2.2. Karakteristik Sistem [3]

2.2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut [7]:

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia dan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia, misalnya sistem perputaran bumi dan sistem buatan manusia adalah sistem yang dirancang oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi dan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya dan sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.2.2 Konsep Dasar Informasi

Informasi (*information*) dapat didefinisikan sebagai berikut: [7]

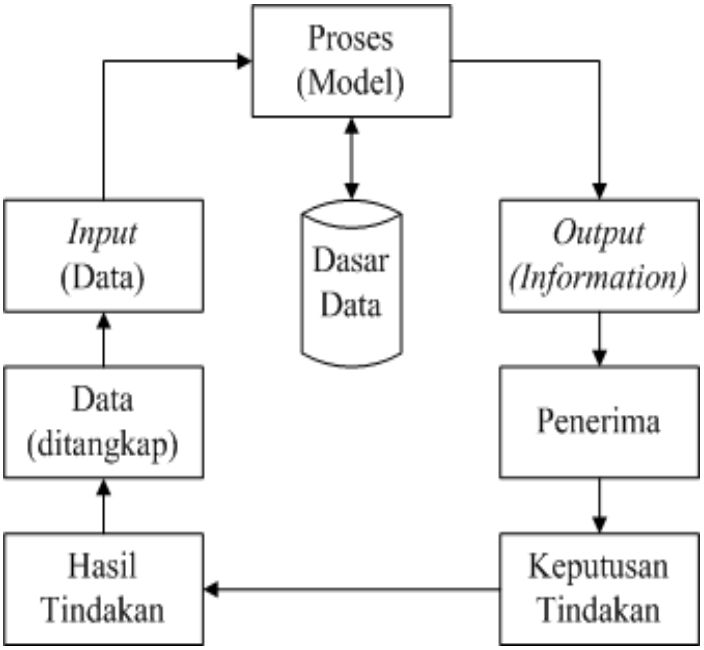
Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data-item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Kesatuan nyata (*fact* dan *entity*) adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

2.2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat berceritera banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi.

Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk siklus. Siklus ini disebut dengan siklus informasi (*information cycle*) atau disebut juga dengan siklus pengolahan data (*data processing cycles*). Siklus informasi dapat dilihat pada gambar 2.3 berikut ini :



Gambar 2.3. Siklus Informasi [7]

2.2.2.2 Kegunaan Informasi

Ada 4 faktor utama yang berhubungan dengan kegunaan informasi :

1. Kualitas informasi (*information quality*)

   Kualitas dari suatu informasi tergantung dari 4 hal, yaitu informasi harus :

   a. Akurat (*accurate*) dan presisi (*precision*)

      Akurat dalam menampilkan informasi dan presisi dalam detail informasi yang diberikan.

   b. Kelengkapan (*completeness*)

      Informasi yang tersedia cukup lengkap untuk setiap *user* dan situasi.

   c. Umur (*age*) dan ketepatan waktu (*timeliness*)

      Umur berarti lamanya waktu dalam meng-*update* informasi dan ketepatan waktu berarti menyediakan informasi secepat mungkin pada saat dibutuhkan sehingga berguna.

   d. Sumber (*source*)

Orang atau organisasi yang menghasilkan informasi.

2. Aksesibilitas informasi (*information accessibility*)
   a. Ketersediaan (*availability*)

      Memberikan informasi kepada yang membutuhkan.

      Informasi dapat diakses oleh yang membutuhkan.

   b. Keabsahan (*admissibility*)

      Keabsahan (boleh atau tidak boleh dipakai) informasi tergantung pada hukum, peraturan atau budaya pada saat tertentu.

3. Presentasi informasi (*information presentation*)
   a. Tingkatan (*level of summarization*)

      Perbandingan antara data asli dengan yang ditampilkan.

      Manipulasi data hingga tingkatan yang sesuai, semakin sederhana semakin baik.

   b. Format

      Bentuk dimana informasi ditampilkan ke *user*.

      Manipulasi data ke dalam bentuk yang sesuai.

4. Keamanan informasi (*information security*)
   a. Batasan akses (*access restriction*)

      Prosedur dan teknik mengontrol *user* yang boleh atau tidak mengakses data pada situasi tertentu.

      Penggunaan password atau teknik lain untuk mencegah *user* yang tidak berhak.

b. Enkripsi (*encryption*)

Konversi data ke bentuk tertentu sehingga tidak dapat dibaca oleh *user* yang tidak berhak.

2.2.2.3 Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. [7]

Sebagian besar informasi tidak dapat persis ditaksir keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitasnya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*.

2.2.3 Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Informasi dapat diperoleh dari sistem informasi (*information systems*) atau disebut juga dengan *processing systems* atau *information processing systems* atau *information-generating systems*. Sistem informasi didefenisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai berikut [7] :

> *Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.*

2.2.3.1 Komponen Sistem Informasi

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian

utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data

   Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut.

6. Blok Kendali

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

Pengelompokan komponen-komponen sistem informasi berbasis komputer adalah sebagai berikut :

1. Perangkat keras (*hardware*)

   Hardware ini merupakan peralatan fisik yang dapat digunakan untuk mengumpulkan, memasukkan, memproses, menyimpan, dan mengeluarkan hasil pengolahan data dalam bentuk informasi.

2. Perangkat lunak (*software*)

   Software adalah kumpulan dari program-program yang digunakan untuk menjalankan aplikasi tertentu pada komputer.

3. Manusia (*brainware*)

   Brainware dalam sistem informasi berperan sebagai pemberi dan pengguna informasi.

4. Prosedur (*procedure*)

   Prosedur adalah rangkaian aktivitas atau kegiatan yang dilakukan secara berulang-ulang dengan cara yang sama.

5. Basis data (*database*)

   Database merupakan kumpulan data-data yang tersimpan di dalam media penyimpanan di suatu perusahaan (arti luas) atau di dalam komputer (arti sempit).

6. Jaringan komunikasi (*communication network*)

   Jaringan telekomunikasi saat ini menghubungkan beberapa daratan dan lautan untuk memindahkan data dalam jumlah besar.

2.2.3.2 Pengembangan Sistem Informasi

Pengembangan sistem (*systems development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang ada. Sewaktu melakukan proses pengembangan sistem, beberapa prinsip harus tidak boleh dilupakan. Prinsip-prinsip ini adalah sebagai berikut : [7]

1. Sistem yang dikembangkan adalah untuk manajemen
2. Sistem yang dikembangkan adalah investasi modal yang besar
3. Sistem yang dikembangkan memerlukan orang yang terdidik
4. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem
5. Proses pengembangan sistem tidak harus urut
6. Jangan takut membatalkan proyek
7. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem

Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan, dan dipelihara. Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah di dalam tahapan tersebut dalam proses pengembangannya.

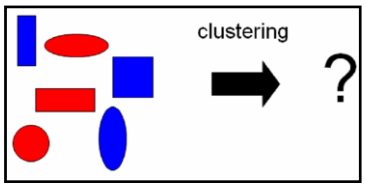
Pengembangan sistem yang digunakan yaitu *classsic life style* atau yang lebih dikenal dengan istilah *waterfall*. Pengembangan sistem menurut A. Ziya Aktas (1987) adalah sebagai berikut :

1. Rekayasa sistem (*system engineering*), merupakan tahap awal dalam pengembangan sistem yaitu dengan menetapkan segala hal yang diperlukan dalam pelaksanaan pengembangan sistem dan menentukan apakah sistem benar-benar dibutuhkan atau tidak. Tahap-tahap yang digunakan yaitu dengan diadakannnya wawancara, observasi, dan studi literatur.
2. Analisis (*analysis*), merupakan tahap menganalisis kebutuhan sistem seperti mendefinisikan kembali masalah, memahami kebutuhan-kebutuhan pemakai dan hambatan-hambatan pada sustu sistem baru, dan membuat model logika dari pemecahan yang direkomendasi. Adapun metode analisis yang digunakan adalah metode analisis terstruktur.
3. Desain (*Design*), yaitu tahap setelah analisis dari siklus pengembangan sistem, pendefinisian dari kebutuhan-kebutuhan fungsional, persiapan untuk rancang bangun implementasi, dan menggambarkan bagaimana suatu sistem dibentuk.
4. Penulisan Program (*Coding*), adalah tahap menterjemahkan hasil analisis ke dalam bahasa pemrograman yang telah ditentukan.

5. Pengujian (*Testing*), tahap dimana melakukan pengujian terhadap sistem yang telah dibangun.
6. Pemeliharaan (*Maintenance*), tahap ini merupakan tahap akhir dimana sistem yang sudah selesai dapat mengalami perubahan atau penambahan sesuai dengan keinginan konsumen.

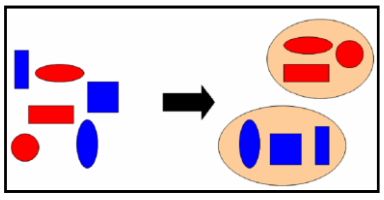
2.2.4 *Cluster*

Cluster adalah sekumpulan objek yang mempunyai "Kesamaan" diantara anggotanya dan memiliki "Ketidaksamaan" dengan objek lain pada *cluster* lainnya, dengan kata lain sebuah *cluster* adalah sekumpulan objek yang digabung bersama karena persamaan atau kedekatannya. *Clustering* adalah proses membuat pengelompokkan sehingga semua anggota dari setiap partisi mempunyai persamaan berdasarkan matrik tertentu. Gambar 2.4 berikut menunjukkan contoh data yang akan dilakukan klasterisasi [9].



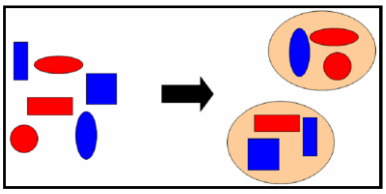
Gambar 2.4 Data Sebelum di Klasterisasi

Jika data dilakukan *clustering* (pengelompokkan) berdasarkan warna, maka pengelompokkannya seperti yang terlihat pada gambar 2.5.



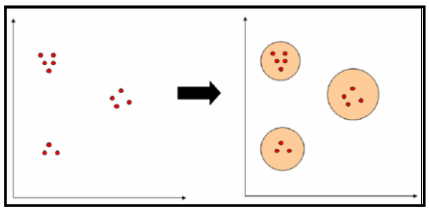
Gambar 2.5 Klasterisasi Berdasarkan Similaritas (Kesamaan) Warna

Jika data dilakukan *clustering* (pengelompokkan) berdasarkan bentuk, maka pengelompokannya dapat dilihat seperti gambar 2.6:



Gambar 2.6 Klasterisasi Berdasarkan Similaritas (Kesamaan) Bentuk

Selain dengan menggunakan similaritas (kesamaan) berdasarkan bentuk dan warna, *clustering* juga bisa dilakukan dengan menggunakan similaritas berdasarkan jarak, artinya data yang memiliki jarak berdekatan akan membentuk satu *cluster*, contohnya seperti dapat dilihat pada gambar 2.7:



Gambar 2.7 Klasterisasi Berdasarkan Similaritas (Kesamaan) Jarak

Ada beberapa perbedaan antara metode klasifikasi dan metode *clustering*, dimana pada dasarnya terdapat tiga poin perbedaan yaitu : data, label dan analisa hasil. Perbedaan tersebut dapat ditabelkan seperti tabel 2.1 berikut:

Tabel 2.1 Perbedaan Klasifikasi dan Klasterisasi

Perbedaan	**Klasifikasi**	**Klasterisasi**
Data	*Supervised*	*Unsupervised*
Label	Ya	Tidak
Analisa Hasil	*Error Ratio*	*Variance*

Data *supervised* pada klasifikasi artinya data melalui pembelajaran terbimbing, sedangkan data *unsupervised* pada klasterisasi artinya data tidak

melalui pembelajaran terbimbing. Analisa hasil pada klasterisasi dinyatakan dengan *variance* yang menunjukkan variansi data dalam satu *cluster*, sedangkan klasifikasi analisa hasil diukur menggunakan rasio kesalahan (*error ratio*). Pada dataset yang digunakan oleh klasifikasi terdapat satu *attribut* (label) yang berfungsi sebagai *attribut* target, sedangkan dataset pada klasterisasi tidak terdapat attribut (label) sebagai attribut target. [9].

2.2.4.1 Karakteristik *Clustering*

Ada beberapa karakteristik dari *cluster*ing, masing-masing akan dijelaskan berikut ini [9]:

1. Partitioning Clustering

A. Disebut juga *exclusive clustering*
B. Setiap data harus termasuk dalam *cluster* tertentu
C. Memungkinkan bagi setiap data yang termasuk *cluster* tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke *cluster* yang lain.

Contoh : *K-Means*, residual analysis.

2. Hierarchical Clustering

A. Setiap data harus masuk ke dalam *cluster* tertentu
B. Suatu data yang masuk kedalam *cluster* tertentu pada suatu tahapan proses, tidak dapat berpindah ke *cluster* lain.

Contoh: *Single Linkage, Centroid Linkage, Complete Linkage*

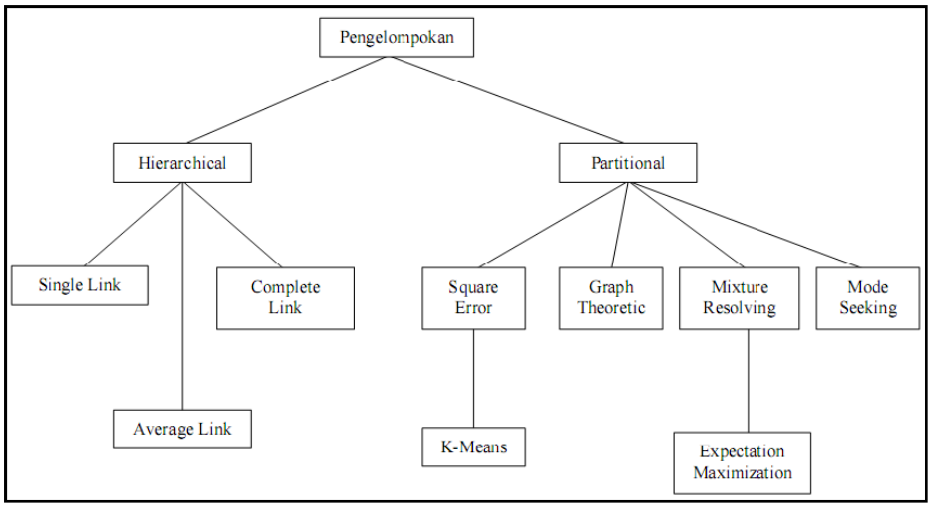
3. Overlapping Clustering

A. Setiap data memungkinkan termasuk ke beberapa *cluster*
B. Data mempunyai nilai keanggotaan (*membership*) pada beberapa *cluster*.

Contoh : *Fuzzy C-means, Gaussian Mixture*

4. Hybrid

Merupakan kombinasi dari karakteristik partitioning, overlapping dan hierarchical.



Gambar 2.8 Karakteristik Klasterisasi

2.2.5 Metode Pengelompokan

Metode pengelompokan pada dasarnya ada dua, yaitu metode pengelompokan Hirarki (*Hirarchical Clustering Method*) dan metode non Hirarki (*Non Hirarchical Clustering Method*). Metode pengelompokan hirarki digunakan apabila belum ada informasi jumlah kelompok yang akan dipilih. Sedangkan metode pengelompokan Non Hirarki bertujuan untuk mengelompokkan *n* objek kedalam *k* kelompok (k<n), dimana nilai *k* telah ditentukan sebelumnya. Salah satu prosedur pengelompokan pada Non Hirarki adalah dengan menggunakan metode *k-means*. Metode ini merupakan metode pengelompokan yang bertujuan untuk mengelompokkan objek sedemikian hingga jarak tiap-tiap objek ke pusat kelompok didalam suatu kelompok adalah minimum. [9].

2.2.6 *Distance Space*

Distance Space berfungsi untuk menghitung jarak antara data dan *centroid*. Ada beberapa macam *distance space* yang sudah diimplementasikan salah satunya adalah : *Euclidean distance space*.[2]

Euclidean sering digunakan karena penghitungan jarak dalam *distance space* ini merupakan jarak terpendek yang bisa didapatkan antara dua titik yang diperhitungkan.

Jarak antara dua titik dapat dihitung dengan cara:

$$d = |x - y| = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

Dimana:

d : jarak

p : dimensi data

x : titik data pertama,

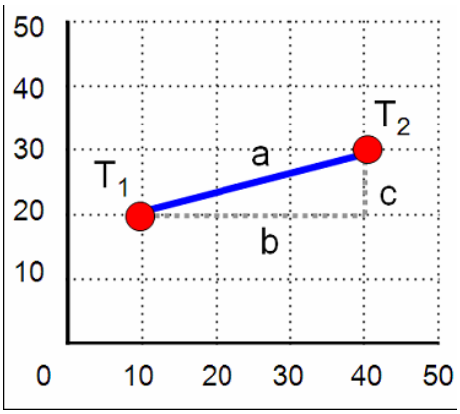
y : titik data kedua,

Dalam *Euclidean* perhitungan yang dilakukan merupakan jarak terpendek antara dua titik.

Jika ada *n* titik pengamatan dengan *p* variabel, maka sebelum dilakukan pengelompokkan data atau objek, terlebih dahulu menentukan ukuran kedekatan sifat antar data. Ukuran data yang bisa digunakan adalah euclidius (*euclidian distance*), antara dua titik dari *p* dimensi pengamatan. Jika antar titik X (x1, x2,...,xn) dan titik Y=(y1,y2,...yn) di tentukan dengan rumus:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 ..... + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

Penghitungan jarak dengan *Euclidian Distance* untuk dua titik seperti diilustrasikan pada gambar 2.9:



$$a^2 = b^2 + c^2$$
$$a = \sqrt{b^2 + c^2}$$
$$a = \sqrt{(40-10)^2 + (30-20)^2}$$
$$a = \sqrt{(30)^2 + (10)^2}$$
$$a = \sqrt{900 + 100}$$
$$a = \sqrt{1000} = 31.628$$

Gambar 2.9 Penghitungan Jarak Dua Titik

Dari penggambaran diatas dapat diartikan bahwa semakin kecil jarak atau nilai *d*, maka semakin besar keserupaan antar objek tersebut.

2.2.7 Analisa *Cluster*

Analisis *cluster* adalah suatu analisis statistik yang bertujuan memisahkan obyek kedalam beberapa kelompok yang mempunyai sifat berbeda antar kelompok yang satu dengan yang lain. Dalam analisis ini tiap-tiap kelompok bersifat homogen antar anggota dalam kelompok atau variasi obyek dalam kelompok yang terbentuk sekecil mungkin.[9]

2.2.7.1 Proses Analisis *Cluster*

Tujuan utama analisis *cluster* menggabungkan objek-objek yang mempunyai kesamaan ke dalam sebuah kelompok atau *cluster*. Untuk mencapai tujuan itu kita harus menjawab tiga pertanyaan, yaitu : [9]

(1) Bagaimana kita mengukur tingkat kesamaan ?

Ada tiga ukuran untuk mengukur kesamaan antar objek, yaitu ukuran korelasi, ukuran jarak, dan ukuran asosiasi.

(2) Bagaimana kita membentuk *cluster* ?

Prosedur yang diterapkan harus dapat mengelompokkan objek-objek yang memiliki kesamaan yang tinggi ke dalam suatu *cluster* yang sama.

(3) Berapa banyak *cluster* yang akan kita bentuk ?

Pada prinsipnya jika jumlah *cluster* berkurang maka homogenitas alam *cluster* secara otomatis akan menurun.

2.2.7.2 Proses Pengambilan Keputusan dalam Analisis *Cluster*

Pengambilan keputusan dengan analisis *cluster* memiliki 6 tahapan, yaitu : menentukan tujuan analisis *cluster*, menentukan desain penelitian analisis *cluster*, menentukan asumsi analisis *cluster*, menurunkan *cluster-cluster* dan memperkirakan *overall fit*, menginterpretasi hasil analisis *cluster*, mengukur tingkat validasi hasil analisis *cluster*. [9]

Langkah 1 : Tujuan Analisis *Cluster*

Tujuan analisis *cluster* secara khusus antara lain :

(1) Pengelompokkan → Analisis *cluster* digunakan dengan tujuan *explanatory* maupun *confirmatory*

(2) Penyederhanaan data → Analisis *cluster* menetapkan struktur dari observasi atau data bukan variabel.

(3) Pengidentifikasian hubungan → Analisis *cluster* dapat menunjukkan ada tidaknya hubungan antar observasi atau obyek dalam analisis

Langkah 2 : Desain Penilitian dalam Analisis *Cluster*

A. Mendeteksi *Outliers*

Dalam melakukan pemilahan obyek kedalam *cluster-cluster*, analisis tidak hanya peka terhadap variabel-veriabel yang tidak sesuai dengan kasus yang diteliti tetapi juga peka terhadap *outliers* (obyek-obyek yang "berbeda" dengan obyek yang lainnya). *Outliers* terjadi karena 2 dua hal, yaitu :

(1) Observasi "menyimpang" yang tidak mewakili populasi

(2) Suatu *undersampling* kelompok-kelompok dalam populasi yang menyebabkan *underrepresentation* kelompok-kelompok dalam sampel.

Dalam kedua kasus tersebut, *outliers* dapat mengubah struktur sebenarnya dari populasi sehingga kita akan memperoleh *cluster-cluster* yang tidak sesuai dengan struktur sebenarnya dari populasi tersebut. Karena itu, pembuangan *outliers* sangat penting dalam analisis ini. *Outliers* dapat dilihat melalui *Profile Diagram. Outliers* adalah obyek-obyek dengan profil-profil yang berbeda, atau *value* yang berbeda dalam satu atau beberapa variabel.

B. Kesamaan Ukuran

Konsep kesamaan yang diperlukan dalam analisis *cluster*. *Interobject Similarity* adalah sebuah ukuran untuk kesesuaian atau kemiripan, diantara objek-objek yang akan dipilah menjadi beberapa *cluster*. *Interobject Similarity* dapat diukur dengan beberapa cara, antara lain : *Correlatioal Measures, Distance Measures,* dan *Association Measures*. Pemilihan metode tergantung pada tujuan dan jenis data. *Correlatioal Measures* dan *Distance Measures* digunakan untuk data dengan tipe *metic,* sedangkan *Association Measures* digunakan bila data bertipe *non-metic*.

C. Standarisasi Data

Sama halnya dengan seleksi kesamaan ukuran, dalam standarisasi data ni peneliti harus menjawab sebuah pertanyaan, yaitu : Apakah data yang tersedia harus distandarisasi? Dalam menjawab pertanyaan ini, penelti harus memperhatikan beberapa masalah, misalnya, jarak nilai dari masing-masing variabel karena perbedaan skala. Secara umum, variabel dengan penyebaran nilai yang tinggi mempunyai dampak yang lebih pada hasil akhir. Karena itu, peneliti diharapkan mengetahui secara lengkap pengukuran dari variabel-variabel. Proses standarisasi dalam analisi *cluster* ada dua, yaitu : standarisasi berdasarkan variabel dan standarisasi berdasarkan observasi.

Langkah 3 : Asumsi-asumsi Analisis *Cluster*

Dalam analisis *cluster*, peneliti harus lebih memperhatikan masalah : seberapa besar sampel mewakili populasi (*representativeness*) dan ada tidaknya *multicollinearity*.

Langkah 4 : Menurunkan *Cluster-Cluster* dan Memperkirakan *Overall Fit*

Peneliti pertama kali harus menentukan *clustering algorithm* yang akan digunakan untuk membentuk *cluster* dan selanjutnya memutuskan berapa *cluster* yang akan dibentuk. Dua hal ini mempunyai implikasi yang substensial tidak hanya pada hasil yang akan diperoleh tetapi juga pada intepretasi hasil tersebut.

Langkah 5 : Interpretasi *Cluster*

Tahap interpretasi meliputi pengujian masing-masing *cluster* dalam terminology macam *cluster* untuk menamai atau memberikan keterangan secara tepat sebagai gambaran sifat dari *cluster*. Ketika memulai proses interpretasi, ada satu ukuran yang sering digunakan yaitu *cluster centroid.* Jika prosedur

pengelompokan dilakukan terhadap data asli, maka ini akan memberikan gambaran yang logic. Tetapi jika data telah distandarisasi atau jika analisis *cluster* dilakukan dengan menggunakan hasil analisis faktor (faktor komponen), peneliti harus mengembalikan skor asli untuk variabel asal dan menghitung rata-rata profiles menggunakan data ini.

Gambaran dan interpretasi *cluster*, memberikan hasil lebih daripada deskriptif. *Pertama*, Metode ini memberikan sebuah rata-rata untuk perkiraan masing-masing *cluster* yang terbentuk sebagaimana yang dikemukakan pada toeri sebelumnya atau pengalaman praktek. *Kedua*, Gambaran *cluster* memberikan jalan untuk membuat perkiraan signifikansi praktis. Peneliti mungkin memerlukan bahwa perbedaan substansi yang ada pada sejumlah variable *cluster* dan penyelesaian *cluster* akan dikembangkan sampai tampak sejumlah perbedaan.

Langkah 6 : Validasi dan Gambaran *Cluster*

Analisis *cluster* agak bersifat subjektif dalam penentuan penyelesaian *cluster* yang optimal, sehingga peneliti seharusnya memberikan perhatian yang besar mengenai validasi dan jaminan tingkat signifikansi pada penyelesaian akhir dari *cluster*. Meskipun tidak ada metode untuk menjamin validitas dan tingkat signifikansi, beberapa pendekatan telah dikemukakan untuk memberikan dasar bagi perkiraan peneliti.

A. Validasi Hasil *Cluster*

Validasi termasuk usaha yang dilakukan oleh peneliti untuk menjamin bahwa hasil *cluster* adalah representatif terhadap populasi secara umum, dan dengan demikian dapat digeneralisasi untuk objek yang lain dan stabil untuk waktu tertentu. Pendekatan langsung dalam hal ini adalah dengan analisis

sample secara terpisah kemudian membandingkan antara hasil *cluster* dengan perkiraan masing-masing *cluster*. Pendekatan ini sering tidak praktis, karena adanya keterbatasan waktu dan biaya atau ketidaktersediaan objek untuk perkalian analisis *cluster*. Dalam hal ini pendekatan yang biasa digunakan adalah dengan membagi sample menjadi dua kelompok. Masing-masing dianalisis *cluster* secara terpisah, kemudian hasinya dibandingkan.

B. Profiling Hasil *Cluster*

Tahap *Profiling* meliputi penggambaran karakteristik masing-masing *cluster* untuk menjelaskan bagaimana mereka bisa berbeda secara relevan pada tiap dimensi. Tipe ini meliputi penggunaan analisis diskriminan. Prosedur dimulai setelah *cluster* ditentukan. Peneliti menggunakan data yang sebelumnya tidak masuk dalam prosedur *cluster* untuk menggambarkan karakteristik masing-masing *cluster*. Meskipun secara teori tidak masuk akal (rasional) dalam perbedaan silang *cluster*, akan tetapi hal ini diperlukan untuk memprediksi validasi taksiran, sehingga minimal penting secara praktek.

2.2.8 *K-Means*

Algoritma *K-Means* adalah Metode *clustering* non *hierarchical* berbasis jarak yang membagi data kedalam *cluster* dan algoritma ini bekerja pada attribut numerik. Algoritma *K-Means* termasuk dalam partitioning *clustering* yang memisahkan data ke *k* daerah bagian yang terpisah. Algoritma *K-Means* sangat terkenal karena kemudahan dan kemampuannya untuk mengklaster data besar dan *outlier* dengan sangat cepat.[2][6]

K-Means merupakan metode klasterisasi yang paling terkenal dan banyak digunakan di berbagai bidang karena sederhana, mudah diimplementasikan, memiliki kemampuan untuk mengklaster data yang besar, mampu menangani data outlier, dan kompleksitas waktunya linear O(nKT) dengan n adalah jumlah dokumen, K adalah jumlah kluster, dan T adalah jumlah iterasi. Dalam algoritma *K-Means*, setiap data harus termasuk ke *cluster* tertentu pada suatu tahapan proses, pada tahapan proses berikutnya dapat berpindah ke *cluster* yang lain. Pada dasarnya penggunaan algoritma *K-Means* dalam melakukan proses *clustering* tergantung dari data yang ada dan konklusi yang ingin dicapai. Untuk itu digunakan algoritma *K-Means* yang didalamnya memuat aturan sebagai berikut:

A. Jumlah *cluster* yang perlu di inputkan
B. Hanya memiliki attribut bertipe numerik

Algoritma *K-Means* pada awalnya mengambil sebagian dari banyaknya komponen dari populasi untuk dijadikan pusat *cluster* awal. Pada step ini pusat *cluster* dipilih secara acak dari sekumpulan populasi data. Berikutnya *K-Means* menguji masing-masing komponen didalam populasi data dan menandai komponen tersebut ke salah satu pusat *cluster* yang telah di definisikan tergantung dari jarak minimum antar komponen dengan tiap-tiap pusat *cluster*. Posisi pusat *cluster* akan dihitung kembali sampai semua komponen data digolongkan kedalam tiap-tiap *cluster* dan terakhir akan terbentuk posisi *cluster* baru.

Algoritma *K-Means* pada dasarnya melakukan 2 proses yakni proses pendeteksian lokasi pusat *cluster* dan proses pencarian anggota dari tiap-tiap *cluster*.

Proses algoritma *K-Means* :

1. Tentukan *k* sebagai jumlah *cluster* yang ingin dibentuk
2. Bangkitkan *k* centroids (titik pusat *cluster*) awal secara random.
3. Hitung jarak setiap data ke masing-masing *centroids*.
4. Setiap data memilih *centroids* yang terdekat
5. Tentukan posisi *centroids* baru dengan cara menghitung nilai rata-rata dari data-data yang terletak pada *centroids* yang sama.
6. Kembali ke langkah 3 jika posisi centroids baru dengan centroids lama tidak sama.

Berdasarkan cara kerjanya, algoritma *K-Means* memiliki karakteristik:

1. *K-Means* sangat cepat dalam proses *clustering*
2. *K-Means* sangat sensitif pada pembangkitan *centroids* awal secara random
3. Memungkinkan suatu *cluster* tidak mempunyai anggota
4. Hasil *clustering* dengan *K-Means* bersifat tidak unik (Selalu berubah-ubah) – terkadang baik, terkadang jelek.

Adapun tujuan dari data *cluster*ing ini adalah untuk meminimalisasikan *objective function* yang diset dalam proses *cluster*ing, yang pada umumnya berusaha meminimalisasikan variasi di dalam suatu *cluster* dan memaksimalisasikan variasi antar *cluster*. Ada dua cara pengalokasian data kembali ke dalam masing-masing *cluster* pada saat proses iterasi *cluster*ing.

K-Means dalam pengalokasian data ke dalam masing-masing *cluster* dapat dilakukan dengan 2 cara yaitu Hard *K-Means* dan Fuzzy K-Means. Perbedaan dari kedua metode tersebut terletak pada asumsi yang dipakai sebagai dasar dari pengalokasian data. Hard disini dalam artian suatu data secara tegas atau pasti

dinyatakan sebagai anggota satu *cluster* tertentu dan tidak menjadi anggota *cluster* yang lain. Sedangkan Fuzzy diartikan masing-masing data mempunyai nilai kemungkinan untuk dapat bergabung ke setiap *cluster* yang ada.

2.2.9 *Hard K-Means*

Pengalokasian kembali data ke dalam masing-masing *cluster* dalam metode *Hard K-Means* didasarkan pada perbandingan jarak antara data dengan *centroid* setiap *cluster* yang ada. Data dialokasikan ulang secara tegas ke *cluster* yang mempunyai *centroid* terdekat dengan data tersebut. Pengalokasian ini dapat dirumuskan sebagai berikut [2][6]

$$a_{ik} = \begin{cases} 1 & d = \min\{D(x_k, v_i)\} \\ 0 & lainnya \end{cases}$$

dimana:

a_{ik} : Keanggotaan data ke-*k* ke *cluster* ke-*i*

v_i : Nilai *centroid cluster* ke-*i*

Metode *Hard K-Means* melakukan proses *cluster*ing dengan mengikuti algoritma sebagai berikut:

a. Tentukan jumlah *cluster*

b. Alokasikan data sesuai dengan jumlah *cluster* yang ditentukan

c. Hitung nilai *centroid* masing-masing *cluster*

d. Alokasikan masing-masing data ke *centroid* terdekat

e. Kembali ke Step c. apabila masih terdapat perpindahan data dari satu *cluster* ke *cluster* yang lain.

Untuk menghitung *centroid cluster* ke-*i*, v_i , digunakan rumus sebagai berikut:

$$v_{ij} = \frac{\sum_{k=1}^{N_i} x_{kj}}{N_i}$$

dimana:

Ni : Jumlah data yang menjadi anggota *cluster* ke-*i*

2.2.10 *Fuzzy K-Means*

Metode *Fuzzy K-Means* (atau lebih sering disebut sebagai *Fuzzy C-Means*) mengalokasikan kembali data ke dalam masing-masing *cluster* dengan memanfaatkan teori *Fuzzy*. Teori ini mengeneralisasikan metode pengalokasian yang bersifat tegas (*hard*) seperti yang digunakan pada metode *Hard K-Means*. Dalam metode *Fuzzy K-Means* dipergunakan variable *membership function*, u_{ik} , yang merujuk pada seberapa besar kemungkinan suatu data bisa menjadi anggota ke dalam suatu *cluster*[2][6]

Pada *Fuzzy K-Means* yang diusulkan oleh Bezdek, diperkenalkan juga suatu variabel *m* yang merupakan *weighting exponent* dari *membership function*. Variabel ini dapat mengubah besaran pengaruh dari *membership function*, u_{ik} , dalam proses *cluster*ing menggunakan metode *Fuzzy K-Means*. *m* mempunyai wilayah nilai *m*>1. Sampai sekarang ini tidak ada ketentuan yang jelas berapa besar nilai *m* yang optimal dalam melakukan proses optimasi suatu permasalahan *cluster*ing. Nilai *m* yang umumnya digunakan adalah 2. *Membership function* untuk suatu data ke suatu *cluster* tertentu dihitung menggunakan rumus sebagai berikut:

$$u_{ik} = \sum_{j=1}^{c} \left(\frac{D(x_k, v_i)}{D(x_k, v_j)} \right)^{\frac{2}{m-1}}$$

dimana:

u_{ik} : *Membership function* data ke-*k* ke *cluster* ke-*i*

v_i : Nilai *centroid cluster* ke-*i*

m : *Weighting Exponent*

Membership function, u_{ik} , mempunyai wilayah nilai $0 \leq u_{ik} \leq 1$.

Data item yang mempunyai tingkat kemungkinan yang lebih tinggi ke suatu kelompok akan mempunyai nilai *membership function* ke kelompok tersebut yang mendekati angka 1 dan ke kelompok yang lain mendekati angka 0. Metode *Fuzzy K-Means* melakukan proses *cluster*ing dengan mengikuti algoritma sebagai berikut :

a. Tentukan jumlah *cluster*
b. Alokasikan data sesuai dengan jumlah *cluster* yang ditentukan
c. Hitung nilai *centroid* dari masing-masing *cluster*
d. Hitung nilai *membership function* masing-masing data ke masing-masing *cluster*
e. Kembali ke Step c. apabila perubahan nilai *membership function* masih di atas nilai *threshold* yang ditentukan.

Untuk menghitung *centroid cluster* ke-*i*, *i c* , digunakan rumus sebagai berikut:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

dimana:

N : Jumlah data

m : *Weighting exponent*

u_{ij} : *Membership function* data ke-*i* ke *cluster* ke-*j*

2.2.11 Perbandingan Algoritma *Clustering*

Perbandingan algoritma *cluster*ing dapat dilihat pada tabel 2.[2] [6]

Table 2.1 Perbandingan Algoritma *Clustering*

Metode	Algoritma	Kelebihan	Kekurangan
K-Means	1. Pilih secara acak *k* vector sebagai centroid 2. repeat 3. tempatkan data (vektor) dalam *cluster* atau centroid terdekat 4. hitung centroid baru dari *cluster* yang terbentuk 5. until centroid tidak berubah lagi	1. Dengan sejumlah besar variabel, *K-Means* mungkin komputasi lebih cepat dari *cluster*ing hirarkis (jika K adalah kecil). 2. *K-Means cluster* dapat menghasilkan lebih ketat daripada *clustering* hirarkis, terutama jika *cluster* globular.	1. Ditemukannya beberapa model *cluster*ing yang berbeda 2. Pemilihan jumlah *cluster* yang paling tepat 3. Kegagalan untuk *converge* 4. awal partisi yang berbeda dapat menghasilkan *cluster* yang berbeda akhir. Hal ini berguna untuk mengulangi program menggunakan sama serta nilai-nilai yang berbeda K, untuk membandingkan hasil yang dicapai.
Fuzzy C-means	1. Pilih jumlah dari *cluster*. 2. Pilih secara acak untuk setiap koefisien poin untuk berada di *cluster*. 3. Ulangi sampai algoritma telah bertemu (yaitu, perubahan koefisien 'antara dua iterasi tidak lebih dari \ varepsilon, diberikan batas sensitivitas): a. Hitung centroid untuk setiap *cluster*, dengan menggunakan rumus .	*1. Unsupervise* 2. Selalu menyatu	1. Waktu yang panjang untuk komputasi 2. Sensitivitas terhadap dugaan awal (kecepatan, minimum lokal) 3. Sensitivitas terhadap suara 4. Satu mengharapkan rendah (atau bahkan tidak ada) tingkat keanggotaan untuk outlier (poin berisik)

Tabel 2.2 Perbandingan Algoritma *Clustering* (Lanjutan)

	b. Untuk setiap titik, hitunglah koefisien yang berada di *cluster*, dengan menggunakan rumus.		
Hierarchical clustering	1. Mulailah dengan menguraikan pengelompokan memiliki tingkat L (0) = 0 dan m urutan nomor = 0. 2. Cari pasangan paling berbeda dari kelompok dalam pengelompokan saat ini, katakanlah pasangan (r), (s), menurut d [(r), (s)] = min d [(i), (j)] dimana minimum lebih dari semua pasangan dari *cluster* di *cluster*ing saat ini. 3. Kenaikan jumlah urutan: m = m 1. Gabung *cluster* (r) dan (s) ke dalam *cluster* tunggal untuk membentuk pengelompokan berikutnya m. Mengatur tingkat ini *cluster*ing untuk L (m) = d [(r), (s)] 4. Update kedekatan matriks, D, dengan menghapus baris dan kolom yang sesuai untuk *cluster* (r) dan (s) dan menambahkan baris dan kolom yang sesuai dengan *cluster* yang baru terbentuk. Kedekatan antara *cluster* baru, dinotasikan	Kadang-kadang berarti bagi data *cluster* di tingkat percobaan bukan pada tingkat gen individu. percobaan seperti yang paling sering digunakan untuk mengidentifikasi kesamaan dalam pola gene-ekspresi secara keseluruhan dalam konteks yang berbeda-tujuan pengobatan rejimen yang sedang untuk mengelompokkan pasien berdasarkan respon mereka molekul-tingkat terhadap perlakuan. Teknik-teknik yang dijelaskan sebelumnya hirarkis sesuai untuk *cluster*ing tersebut, yang didasarkan pada statistik perbandingan berpasangan dari scatterplots lengkap daripada sekuens gen individu. Data direpresentasikan sebagai matriks scatterplots, akhirnya direduksi menjadi matriks koefisien korelasi. Koefisien korelasi yang kemudian digunakan untuk membangun	1. Sekala tidak baik: kompleksitas waktu minimal O (n2), di mana n adalah jumlah objek total; 2. Tidak pernah dapat membatalkan apa yang telah dilakukan sebelumnya.

Tabel 2.2 Perbandingan Algoritma *Clustering* (Lanjutan)

	(r, s) dan *cluster* lama (k) didefinisikan dengan cara ini: d [(k), (r, s)] = min d [(k), (r)], d [(k), (s)] 5. Jika semua benda berada di salah satu *cluster*, berhenti. Lain, lanjutkan ke langkah 2.	dendrogram dua dimensi dengan cara yang sama seperti pada percobaan gen-*cluster* sebelumnya dijelaskan.	
QT *cluster*ing	1. Pengguna memilih diameter maksimum untuk *cluster*. 2. Membangun *cluster* kandidat untuk setiap titik dengan memasukkan titik terdekat, terdekat berikutnya, dan seterusnya, sampai diameter dari *cluster* melampaui ambang batas. 3. Simpan kandidat *cluster* dengan poin terbanyak sebagai *cluster* pertama benar, dan menghapus semua poin dalam *cluster* dari pertimbangan lebih lanjut. Harus menjelaskan apa yang terjadi jika lebih dari 1 *cluster* memiliki jumlah maksimum poin? 4. Recurse dengan kumpulan poin penurunan.	1. Jaminan Kualitas: *cluster* hanya yang lulus batas kualitas didefinisikan pengguna akan dikembalikan. 2. Jumlah dari *cluster* adalah tidak ditentukan a priori: tidak mengharuskan pengguna untuk menentukan jumlah *cluster* dalam lanjutan (seperti peta mengorganisir diri atau k-means). 3. Seluruh *cluster* yang mungkin dianggap: *cluster* kandidat dihasilkan sehubungan dengan setiap gen dan diuji sesuai urutan besarnya terhadap kriteria kualitas.	Komputasi intensif / Sisa Mengkonsumsi: Ukuran Minimum meningkatkan *Cluster*, mengurangi Korelasi Minimum, atau meningkatkan jumlah gen pada gen yang dipilih daftar dapat sangat meningkatkan waktu komputasi.
K-Median	1. Menginisialisasi: k secara acak memilih n titik data sebagai mediods 2. Associate setiap titik data ke medoid terdekat. ("Paling dekat" di sini	Tidak sensitif terhadap outlier, namun membutuhkan waktu yang lebih lama untuk mencari centroid (median)	– Jika ada Anda memiliki set besar angka, itu akan memakan waktu untuk setiap tempat di urutan besarnya

Tabel 2.2 Perbandingan Algoritma *Clustering* (Lanjutan)

	didefinisikan menggunakan metrik jarak yang valid, paling sering jarak Euclid, Manhattan jarak atau jarak Minkowski) 3. Untuk setiap mediod m 1. Untuk setiap titik data non-medoid o 1. Swap m dan o dan menghitung total biaya konfigurasi 4. Pilih konfigurasi dengan biaya terendah. 5. ulangi langkah 2-5 sampai tidak ada perubahan medoid tersebut.		– Jika salah satu nomor dekat bagian tengah dari distribusi bergerak sedikit pun, maka median akan mengubah, tidak seperti mean, yang relatif tidak terpengaruh oleh perubahan di salah satu nomor pusat.

2.2.12 Basis Data (*Database*)

Landasan teori yang menjelaskan tentang *database* meliputi pengertian data, pengolah data, skilus pengolahan data, konsep basis data, pengertian basis data, tujuan basis data, komponen basis data.[5]

2.2.12.1 Pengertian Data

Data merupakan ”fakta atau keterangan yang belum mempunyai arti atau nilai, serta data dapat dijadikan kajian analisis atau kesimpulan. Data biasanya terdiri dari beberapa elemen data (data item). Elemen data adalah unit terkecil dari data yang ada artinya bagi pengguna *(user).*

2.2.12.2 Pengolahan Data

Pengolahan data dengan komputer terkenal dengan nama pengolahan data elektronik. Data adalah kumpulan kejadian yang diangkat dari suatu kenyataan. Data dapat berupa angka-angka, huruf-huruf atau simbol-simbol khusus atau gabungan darinya. Pengolahan data adalah manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti, berupa suatu informasi.

2.2.12.3 Siklus Pengolahan Data

Suatu proses pengolahan data terdiri dari tiga tahapan dasar yang disebut dengan siklus pengolahan data (*data processing cycle*) yaitu *input*, *processing* dan *output*.

1. *Input*, tahap ini merupakan proses memasukkan data ke dalam proses komputer lewat alat *input* (*input device*).
2. *Process*, tahap ini merupakan proses pengolahan dari data yang sudah dimasukkan yang dilakukan oleh alat pemroses (*processing data*), yang

dapat berupa proses menghitung, membandingkan, mengklasifikasikan, mengurutkan, mengendalikan atau mencari di *storage*.

3. *Output*, tahp ini merupakan proses menghasilkan *output* dari hasil pengolahan data ke alat *output* (*output device*), yaitu berupa informasi.

2.2.12.4 Konsep Basis Data

Basis data (*database)* dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis data menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkungan perusahaan atau instansi-instansi. Penerapan basis data dalam sistem informasi disebut sistem basis data (*database system).*

2.2.12.5 Pengertian Basis Data

Basis Data terdiri dari atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya.

"Basis data dapat diartikan sebagai himpunan atau sekumpulan data, bisa berupa tabel atau file yang saling berhubungan dan disimpan dalam media penyimpanan elektronis tanpa pengulangan (*redundansi).*"*[5]

Jadi dapat disimpulkan bahwa basis data merupakan kumpulan data (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai

kebutuhan. Atau bisa diartikan sebagai kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.2.12.6 Tujuan Basis Data

Basis data (*database*) pada prinsipnya mempunyai tujuan awal dan utama dalam pengelolaan data dalam sebuah basis data agar memperoleh atau menemukan kembali data yang dicari dengan mudah dan cepat. Disamping itu, pemanfaatan basis data untuk pengelolaan data, juga memiliki tujuan lain seperti berikut :

1. Kecepatan dan kemudahan (*Speed)*
2. Efisiensi ruang penyimpanan (*Space)*
3. Keakuratan (*Accuracy)*
4. Ketersediaan (*Availability)*
5. Kelengkapan (*Completeness)*
6. Keamanan (*Security)*
7. Kebersamaan Pemakaian (*Sharability)*

2.2.12.7 Komponen Pendukung Basis Data

Basis data hanya sebuah objek yang pasif/mati. Basis data tidak akan pernah berguna jika tidak ada penggeraknya, yang menjadi pengelola atau penggeraknya secara langsung adalah program atau aplikasi (*software)*. Gabungan keduanya (basis data dan pengelolanya) menghasilkan sebuah sistem. Karena itu, secara umum sistem basis data merupakan sistem yang terdiri atas kumpulan file (tabel) yang saling berhubungan dan sekumpulan program *Database Management System* (DBMS) yang memungkinkan beberapa pemakai (program lain) untuk

mengakses dan memanipulasi file-file (tabel-tabel) tersebut. Sistem basis data memiliki beberapa komponen pendukung diantaranya :

a. Perangkat Keras

Perangkat keras yang biasanya terdapat dalam sebuah sistem basis data adalah

1. Komputer (satu untuk sistem yang *stand-alone* atau lebih untuk sistem jaringan).
2. Memori sekunder yang *on-line* (*harddisk).*
3. Memori sekunder yang *off-line* (*tape* atau *removeable disk)* untuk keperluan *backup* data.
4. Media/perangkat komunikasi (untuk sistem jaringan).

b. Sistem Operasi (*Operating Sistem)*

Secara sederhana, sistem operasi merupakan program yang mengaktifkan/memfungsikan sistem komputer, mengendalikan seluruh sumber daya (*resource)* dalam komputer dan melakukan operasi-operasi dasar dalam komputer (operasi I/O, pengelolaan file, dan lain-lain). Sejumlah sistem operasi yang banyak digunakan seperti : *MS-DOS, MS-Windows 3.1, MS-Windows XP, MS-Windows Vista (2007), UNIX*, dan lain-lain. Program pengelola basis data hanya dapat aktif jika sistem operasi yang dikehendaki (sesuai) telah aktif.

c. Basis Data (*Database)*

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi/memiliki sejumlah objek basis data (seperti file/tabel, indeks, dan lain-lain). Disamping berisi/menyimpan data, setiap basis data juga

mengandung definisi struktur (baik untuk basis data maupun objek-objeknya secara detail).

d. Sistem Pengelola Basis Data (*Database Management System/DBMS*)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/spesifik. Perangkat lunak inilah yang akan menentukan bagaimana data organisasi, disimpan, diubah dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan konsistensi data, dan sebagainya.

Perangkat lunak yang termasuk DBMS seperti *dBase III+*, *dBaseIV, FoxBase*, *Rbase, MS-Access* dan *Borland-Paradox* (untuk kelas sederhana) atau *Borland-Interbase, MS-SQLServer, CA-Open Ingres, Oracle, Informix* dan *Sybase* (untuk kelas kompleks/berat).

e. Pemakai (*User)*

Ada beberapa jenis/tipe pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem :

1. *Programmer* Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML), yang disertakan dalam program yang ditulis dalam bahasa pemrograman induk (seperti C, Pascal, Cobol, dan lain-lain).

2. *User* Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan *query* (untuk akses data) dengan bahasa *query* yang telah disediakan oleh DBMS.

3. *User* Umum (*End User Naive User)*

   Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen (*executable program)* yang telah ditulis/disediakan sebelumnya.

4. *User* Khusus (*Specialized User)*

   Pemakai yang menulis aplikasi basis data non konvensional tetapi untuk keperluan-keperluan khusus, seperti untuk aplikasi AI, Sistem Pakar, Pengolahan Citra, dan lain-lain, yang bisa saja mengakses basis data dengan /tanpa DBMS yang bersangkutan.

   Untuk sebuah basis data yang *stand-alone*, maka pada suatu saat hanya ada satu pemakai yang dapat bekerja. Sedangkan untuk sistem basis data dalam jaringan, maka pada suatu saat ada banyak pemakai yang dapat berhubungan (menggunakan) basis data yang sama.

f. Aplikasi (Perangkat Lunak) Lain

   Aplikasi (perangkat lunak) lain ini bersifat opsional, artinya ada tidaknya tergantung kebutuhan. *Database Management System* (DBMS) yang digunakan lebih berperan dalam pengorganisasian data dalam basis data, sementara bagi pemakai basis data (khususnya yang menjadi *end-user/naïve-user)* dapat disediakan program khusus lain untuk melakukan pengisian,

pengubahan dan pengambilan data. Program ini ada yang sudah disediakan bersama dengan DBMS-nya, ada juga yang harus dibuat sendiri dengan menggunakan aplikasi lain yang khusus untuk itu (*development tools).*

2.2.13 *Database Management System (DBMS*)

Diperlukan suatu sistem untuk diintegrasikan data file ke dalam suatu file sehingga bisa melayani berbagai *user* yang berbeda. Perangkat keras dan lunak serta prosedur yang mengelola *database* manajemen sistem. DBMS memungkinkan untuk membentuk dan meremajakan file-file, memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporan-laporan [4].

Fungsi yang penting dari DBMS adalah sebagai berikut :

1. Menyediakan sistem akses cepat.
2. Mengurangi kerangkapan data atau *redundancy* data.
3. Memungkinkan adanya *updating* secara bersamaan.
4. Menyediakan sistem yang memungkinkan dilakukannya pengembangan *database*.
5. Memberikan perlindungan dari pihak pemakai tidak berhak.

2.2.14 Diagram Alir Dokumen / *Flowmap*

Bagan alir dokumen atau disebut juga bagan alir formulir merupakan bagan alir yang menujukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen menggambarkan aliran dokumen dan informasi arus antar area pertanggungjawaban di dalam sebuah organisasi. Secara rinci bagan alir ini menunjukkan dari mana dokumen tersebut berasal, distribusinya, tujuan digunakannya dokumen tersebut dan lain-lain. Bagan alir ini bermanfaat untuk menganalisis kecukupan prosedur pengawasan dalam sebuah sistem.[10]

2.2.15 Alat-alat Pemodelan Sistem

Pemodelan sistem merupakan hal yang penting bagi kelangsungan sistem itu sendiri. Pemodelan sistem adalah suatu upaya untuk menjaga efektivitas sistem dalam memenuhi kebutuhan pengguna sistem. Pemodelan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang sudah ada.[10]

2.2.15.1 *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram atau dikenal dengan diagram E-R secara grafis menggambarkan isi sebuah *database*. Diagram ini memiliki dua komponen utama yaitu *entity* dan relasi. Untuk melambangkan fungsi di atas maka digunakan simbol-simbol.

Adapun elemen-elemen *Entity Relationship Diagram* adalah sebagai berikut :

1. *Entity*

   Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu : orang, benda, lokasi kejadian (terdapat unsur waktu didalamnya).

2. Relasi

   Pada E-R diagram, relasi dapat digambarkan dalam sebuah bentuk belah ketupat. Relasi adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (relasi) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya. Penggambaran

hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

3. Derajat Relasi

   Derajat relasi adalah jumlah entitas yang berpartisipasi dalam suatu relasi.

4. Atribut

   Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap relasi. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun relasi, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan relasi.

5. Kardinalitas

   Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berrelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas, kardinalitas relasi menunjuk pada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu :

   a. One to One

      Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

   b. *One to Many atau Many to One*

      Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada

entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

2.2.15.2 Diagram Konteks / *Context Diagram*

Diagram konteks menggambarkan aplikasi dalam satu lingkaran dan hubungan dengan entitas luar. Dimana lingkaran tersebut menggambarkan keseluruhan proses dalam aplikasi. Dalam penggambaran ini, sistem dianggap sebagai sebuah objek yang tidak dijelaskan secara rinci, karena yang ditekankan adalah interaksi sistem dengan lingkungan yang mengaksesnya. [9]

2.2.15.3 *Data Flow Diagram (DFD)*

Data flow diagram adalah diagram sistem yang menggambarkan cara kerja aplikasi secara *logic*. Mulai dari tingkat paling tinggi sampai dengan tingkat paling rendah. Pada perancangan ini terdiri dari perancangan awal *(preliminary design*) dan perancangan rinci *(detailed design)* sesuai dengan tahap-tahap rekayasa perangkat lunak. Adapun penjelasan dari perancangan awal adalah perancangan sistem yang menggambarkan tentang hubungan antara sistem dengan lingkungan luar sistem. [9]

Hubungan ini dapat digambarkan dengan menggunakan diagram konteks, sedangkan perancangan rinci adalah perancangan sistem yang menggambarkan

tentang proses yang terjadi pada sistem serta arus data yang mengalir antar proses. DFD merupakan alat yang digunakan pada metode pengembangan sistem yang terstruktur, DFD ini menggambarkan arus data di dalam sistem yang terstruktur dan jelas serta merupakan dokumentasi sistem yang baik.

2.2.15.4 Kamus Data *(Data Dictionary)*

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dapat mendefinisikan data yang mengalir pada sistem dengan lengkap. Kamus data dapat digunakan pada tahap analisa dan perancangan sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang masukan, merancang laporan-laporan dan *database.* [9]

Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem.

Kamus data hampir selalu diimplementasikan sebagai bagian dari sebuah piranti desain dan analisis terstruktur. Sebagian besar kamus data berisi format sebagai berikut :

1. *Name*

   Nama sebenarnya dari data atau item kontrol, penyimpanan data atau entitas eksternal.

2. *Aliasi*

Nama lain yang digunakan untuk *entri* pertama.

3. *Where Used / How Used*

Suatu daftar dari proses yang menggunakan data atau item kontrol dan bagaimana dia digunakan.

4. *Content Description*

Suatu notasi untuk mendeskripsikan suatu isi data.

5. *Supplementary Information*

Informasi lain mengenai tipe data, harga perset dll.

Dengan adanya kamus data, didapat definisi-definisi dari bentuk-bentuk yang tidak dimengerti dalam DFD yaitu aliran data, *file*, proses dan elemen-elemen data. Arus data pada DFD bersifat global, hanya ditujukan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari arus data, secara lebih lengkap dapat dilihat di kamus data.

2.2.16 Borland Delphi

Borland delphi adalah paket bahasa pemrograman yang bekerja dalam Sistem Operasi Windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih.[1]

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk

merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis Windows. [6]

Kelebihan-kelebihan yang dapat diambil ketika seorang pengembang perangkat lunak menggunakan Borland Delphi adalah :

1. Delphi mendukung pemrograman berorientasi objek (*Object Oriented Programming*).

2. Hasil dari proses kompilasi berupa sebuah file yang dapat dieksekusi (*executable file*) sehingga mempermudah dalam pendistribusian program dan mengurangi banyaknya *file* pendukung.

3. Delphi menyediakan banyak sekali komponen yang dapat digunakan. Selain itu banyak juga komponen yang bersumber dari pihak ketiga yang biasanya disertai dengan dokumentasi, *source code* dan lain-lain. Komponen dari pihak ketiga bisa yang komersil atau *free*.

4. Mendukung banyak *database server* (*MySQL, SQL Server, Interbase, Oracle* dll) sehingga dapat mempermudah dalam membuat aplikasi *database*.

5. Borland Delphi menyediakan fasilitas yang luas mulai dari fungsi membuat *form* hingga untuk menggunakan format *file* berbasis data yang popular seperti *Dbase* dan *Paradoks*.

6. Dalam Borland Delphi *template* aplikasi dan *template* format yang dapat digunakan untuk membuat semua aplikasi dengan lebih cepat.

2.2.17 *Database* MySQL

MySQL adalah suatu perangkat lunak *database* relasi (*Relational Database Management system* atau RDBMS), seperti halnya ORACLE, Postgresql, *MSSQL* dan sebagainya. Jangan disalah-artikan dengan SQL. SQL (*Structured Query Language*) sendiri didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa (program) yang digunakan untuk mengelola suatu *database*.[4]

Kepopuleran *MySQL* dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil. *Database MySQL* merupakan *database* yang menjanjikan sebagai *alternative* pilihan *database* yang dapat digunakan untuk sistem *database personal* atau organisasi.

Di bawah ini contoh membuat *database*, tabel dan sintak *MySQL* sebagai berikut :

```
mysql> create database spk_kenaikan_pegawai;
mysql> use spk_kenaikan_pegawai;
mysql> create table syarat_pemilihan (
        kode_KP smallint primary key,
        nama_KP varchar(50));
mysql>insert into tkriteria_pemilihan (Kode_KP, Nama_KP)
values ('KP002', 'Sikap Kerja');
insert into tkriteria_pemilihan (Kode_KP, Nama_KP)
values ('KP003', 'Sikap Perilaku');
```