

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Telah ada beberapa penelitian terdahulu yang dilakukan terkait dengan pemilihan ketua OSIS.

Penelitian yang dilakukan oleh Ikhwani Y (2018), yang berjudul Analisis dan Rancangan Sistem E-Voting Pemilihan Ketua OSIS. Dalam melakukan kegiatannya pemilihan ketua OSIS pada sekolah masih sering ditemukan menggunakan media penyampaian informasi yang sederhana yaitu berupa hanya lembaran kertas untuk penyampaian visi dan misi, kemudian ketika pemilihan pun masih menggunakan lembar kertas untuk memilih calon sehingga masih belum maksimal. Dengan memanfaatkan teknologi komputerisasi dengan menggunakan aplikasi pemilihan ketua OSIS (e-voting) berbasis web bisa membuat kegiatan pemilihan ketua OSIS lebih mudah dalam penyampaian informasi, lebih cepat dalam melakukan perhitungan suara, serta membuat lebih hemat terhadap kertas sehingga hasil yang didapat bisa efektif dan efisien. Sehingga perlu melakukan penelitian untuk menganalisis, merancang dan implementasi Sistem E Voting Pemilihan Ketua OSIS dengan berbasis Web Pada Sekolah.

Penelitian yang dilakukan oleh Crystal LYS (2017), dengan judul Sistem Informasi Pemilihan Calon Ketua OSIS di SMA Pawayan Daha. Dalam pengembangan sistem yang dilakukan dalam penelitian ini adalah Metode

Waterfall. Berdasarkan landasan pengembangan aplikasi model waterfall maka penelitian diawali dengan analisis perancangan sistem, kemudian mendesain sistem, membuat kode sistem, dan diakhiri dengan pengujian. Program yang dibuat menggunakan PHP dengan database MySQL. Dalam tahap ini penulis melakukan survey pada lokasi penelitian guna memperoleh data-data yang dibutuhkan dalam melakukan penelitian. Proses yang dilakukan dalam tahap ini yaitu teknik pengumpulan data yaitu wawancara, observasi untuk menentukan ruang lingkup sistem dan membuat jadwal kegiatan.

Penelitian yang dilakukan oleh Anis A Trisnani, dkk (2018) dengan judul Sistem Pendukung Keputusan Pemilihan Karyawan Berprestasi Menerapkan Metode Vise Kriterijumska Optimizajica I Kompromisno Resenje (VIKOR). Berdasarkan hasil penelitian yang telah dilaksanakan bahwa Sistem Pendukung Keputusan (SPK) penentuan karyawan berprestasi dapat dihitung berdasarkan perhitungan dari bobot kriteria masing-masing, sehingga dapat memilih karyawan berprestasi di dalam perusahaan secara cepat menggunakan metode Vise Kriterijumska Optimizajica I Kompromisno Resenje (VIKOR).

II.2. Landasan Teori

II.2.1. Data

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya

yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep (Eka Iswandy, 2015).

II.2.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan (*Decision Support System*) adalah sistem berbasis komputer yang interaktif dalam membantu pengambil keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur. *Decision Support System* (DSS) dengan didukung oleh sebuah sistem informasi berbasis komputer dapat membantu seseorang meningkatkan kinerjanya dalam pengambilan keputusan. Seorang manajer di suatu perusahaan dapat memecahkan masalah semi terstruktur, sehingga manajer dan komputer harus bekerja sama sebagai tim pemecah masalah dalam memecahkan masalah yang berada di area semi terstruktur (Heny Pratiwi, 2016 : 4).

Sedangkan menurut Moore dan Chang, Sistem Pendukung Keputusan adalah sistem yang dapat dikembangkan, mampu mendukung analisis data dan pemodelan keputusan, berorientasi pada perencanaan masa mendatang, serta tidak bisa direncanakan *interval* (periode) waktu pemakaiannya.

Bonezek, Hosapple dan Whinston mendefinisikan sistem pendukung keputusan sebagai suatu sistem yang berbasiskan komputer yang terdiri dari 3 komponen yang berinteraksi satu dengan yang lainnya (Chaulina Alfianti Oktavia, et al.,2015) yaitu :

1. *Language system*, adalah suatu mekanisme untuk menjembatani (*interface*) pemakai dan komponen lainnya.

2. *Knowledge system*, adalah repositori pengetahuan yang berhubungan dengan masalah tertentu baik berupa data maupun prosedur.
3. *Problem processing system*, adalah sebagai penghubung kedua komponen lainnya, berisi satu atau beberapa kemampuan manipulasi atau menyediakan masalah secara umum, yang diperlukan dalam pengambilan keputusan.

II.2.2.1. Konsep Pengambilan Keputusan

Konsep SPK pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael Scott Morton yang selanjutnya dikenal dengan istilah “*Management Decision System*”. Konsep SPK merupakan sebuah sistem interaktif berbasis komputer yang membantu pembuatan keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang bersifat tidak terstruktur dan semi terstruktur. SPK dirancang untuk menunjang seluruh tahapan pembuatan keputusan, yang dimulai dari tahapan mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan sampai pada kegiatan mengevaluasi pemilihan alternative.

II.2.2.2. Tujuan Sistem Penunjang Keputusan

Sistem Penunjang Keputusan mempunyai tiga tujuan yang akan dicapai yaitu :

1. Membantu manajer membuat keputusan untuk memecahkan masalah semistruktur.

2. Mendukung penilaian manajer bukan mencoba menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan manajer dari pada efisiensinya (Iswandy Eka, 2015 : 74).

II.2.2.3. Karakteristik Dan Kemampuan Sistem Penunjang Keputusan (SPK)

Karakteristik dan kemampuan sebuah sistem penunjang keputusan sebagai berikut :

1. Sistem pendukung keputusan menyediakan dukungan untuk pengambilan keputusan utamanya pada keadaan-keadaan semistruktur dan tidak terstruktur dengan menggabungkan penilaian manusia dan informasi komputerisasi.
2. Menyediakan dukungan untuk tingkat manajerial mulai dari eksekutif sampai manejer.
3. Menyediakan dukungan untuk kelompok individu, problem-problem yang kurang terstruktur memerlukan keterlibatan beberapa individu dari departemen-departemen yang lain.
4. Sistem pendukung keputusan menyediakan dukungan kepada independen atau keputusan yang berlanjut.
5. Sistem pendukung keputusan memberikan dukungan kepada semua fase, dalam proses pembuatan keputusan *intelligence, design, choise*, dan implementasi.
6. Sistem pendukung keputusan mendukung banyak proses dan gaya pengambilan keputusan.

7. Sistem pendukung keputusan adaptif terhadap waktu, pembuatan keputusan harus kreatif bisa menghadapi perubahan-perubahan kondisi secara cepat dan merubah sistem pendukung keputusan harus fleksibel sehingga pengguna dapat menambah, menghapus, mengkombinasi, merubah, dan mengatur kembali terhadap elemen-elemen dasar.
8. Sistem pendukung keputusan mudah digunakan. Pengguna merasa berada dirumah saat bekerja dengan *system*, seperti *user friendly*, fleksibilitas, kemampuan penggunaan grafik yang tinggi dan bahasa untuk berinteraksi dengan mesin seperti menggunakan bahasa inggris maka akan menaikkan efektifitas dari sistem pendukung keputusan.
9. Sistem pendukung keputusan menaikkan efektifitas pembuatan keputusan baik dalam hal ketepatan waktu dan kualitas bukan pada biaya pembuatan keputusan atau biaya pemakaian waktu komputer.
10. Pembuat keputusan dapat mengontrol terhadap tahapan-tahapan pembuatan keputusan seperti pada tahap *intelegence, choice and implementation* dan sistem pendukung keputusan diarahkan untuk mendukung pada pembuat keputusan bukan menggantikan posisinya.
11. Memungkinkan pengguna akhir dapat membangun sistem sendiri yang sederhana. Sistem yang besar dapat dibangun dengan bantuan dari spesialis sistem informasi.
12. Sistem pendukung keputusan menggunakan model-model standar atau buatan pengguna untuk menganalisa keadaan-keadaan keputusan. Kemampuan *modeling* (pemodelan) memungkinkan bereksperimen

dengan strategi yang berbeda-beda dibawah konfigurasi yang berbeda-beda pula (Iswany Eka, 2015 : 74).

II.2.3. Metode Vise Kriterijumska Optimizacija I Kompromisno Resenje (VIKOR)

Metode VIKOR diperkenalkan pertama kali oleh Opricovic dan Tzeng, metode dapat didefinisikan sebagai multi kriteria sistem kompleks yang dapat dilihat pada ranking dan pemilihan dari serangkaian alternatif berdasarkan kriteria. Setiap alternatif dievaluasikan sesuai dengan fungsi kriteria. Pemberian peringkat dapat dilakukan dengan membandingkan dan mengukur alternatif-alternatif. Metode Vise Kriterijumska Optimizacija I Kompromisno Resenje (VIKOR) bertujuan untuk mendapatkan hasil perankingan alternatif yang mendekati solusi ideal dengan mengusulkan solusi kompromi. Metode VIKOR sangat berguna pada situasi dimana pengambil keputusan tidak memiliki kemampuan untuk menentukan pilihan pada saat desain sebuah sistem dimulai. (Tumanggor H, dkk, 2018)

Vise Kriterijumska Optimizacija I Kompromisno Resenje (VIKOR) berarti multi-criteria optimization and compromise solution (optimasi multi kriteria dan solusi kompromis), merupakan salah satu dari sekian banyak teknik MCDM. VIKOR diperkenalkan pertama kali oleh Serafim Opricovic pada tahun 1998. VIKOR merupakan metode MCDM yang dapat melakukan perankingan.

Langkah - langkah perhitungan dengan metode VIKOR, sebagai berikut:

1. Melakukan normalisasi menggunakan rumus sebagai berikut:

$$R_{ij} = (X_{j+} - X_{ij} X_{j+} - X_{j-}) \dots\dots\dots (1)$$

Dimana R_{ij} dan X_{ij} ($i=1,2,3,\dots,m$ dan $j=1,2,3,\dots,n$) adalah elemen dari matriks pengambilan keputusan (alternatif iterhadap kriteria j) dan $X_{+ j}$ adalah elemen terbaik dari kriteria j , X_{j-} adalah elemen terbaik dari kriteria j .

2. Menghitung nilai S dan R menggunakan rumus:

$$S_i = \sum W_j(X_{j+} - X_{ij} X_{j+} - X_{j-}) \dots\dots\dots (2)$$

$$\text{dan } R_i = \text{Max } j[w_j(X_{j+} - X_{ij} X_{j+} - X_{j-})] \dots\dots\dots (3)$$

Dimana W_j adalah bobot dari tiap kriteria j .

3. Menentukan nilai indeks

$$Q_i = [S_i - S^+ S^- - S^+]V + [R_i - R^+ R^- - R^+](1-V)\dots\dots\dots(4)$$

Dimana $S^- = \max S_i$, $S^+ = \min S_i$ dan $R^- = \max R_i$, $R^+ = \min R_i$ dan $v = 0,5$.

4. Solusi alternatif peringkat terbaik berdasarkan dengan nilai Q minimum menjadi peringkat terbaik dengan syarat:

$$Q(A(2)) - Q(A(1)) \geq DQ \dots\dots\dots(5)$$

Dimana $A(2)$ = alternatif dengan urutan kedua pada perengkingan Q dan

$A(1)$ = alternatif dengan urutan terbaik pada perengkingan Q sedangkan

$DQ = 1 - (m-1)$,dimana m merupakan jumlah alternatif.

Alternatif $A(1)$ harus berada pada rangking terbaik pada S dan/atau R .

II.2.4. Osis

Osis adalah satu-satunya wadah organisasi siswa yang sah di sekolah. Oleh karena itu setiap sekolah wajib membentuk Organisasi Siswa Intra Sekolah (OSIS), yang tidak mempunyai hubungan organisatoris dengan OSIS di sekolah lain dan tidak menjadi bagian/alat dari organisasi lain yang ada di luar sekolah. Sedangkan apabila OSIS dipandang sebagai suatu sistem, berarti OSIS sebagai tempat kehidupan berkelompok siswa yang bekerjasama untuk mencapai tujuan bersama. Dalam hal ini OSIS dipandang sebagai suatu sistem, dimana sekumpulan para siswa mengadakan koordinasi dalam upaya menciptakan suatu organisasi yang mampu mencapai tujuan. Oleh karena OSIS Sebagai suatu sistem ditandai beberapa ciri pokok, yaitu :

1. Berorientasi pada tujuan
2. Memiliki susunan kehidupan berkelompok
3. Memiliki sejumlah peranan
4. Terkoordinasi
5. Berkelanjutan dalam waktu tertentu. (Zulfikar dan dkk, 2016).

II.2.5. Microsoft Visual Basic

Visual Basic adalah program yang digunakan untuk membuat aplikasi berbasis *Microsoft Windows* secara cepat dan mudah. *Visual basic* menyediakan *tool* untuk membuat aplikasi yang sederhana sampai aplikasi yang kompleks atau rumit baik untuk keperluan pribadi atau untuk keperluan perusahaan atau instansi dengan sistem yang lebih besar. *Microsoft Visual Basic* merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE)

visual untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman.

Dapat disimpulkan bahwa visual basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.

II.2.6. Database

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan, karakter, atau simbol). Basis data adalah sistem yang terkomputerisasi dengan tujuan melakukan pemeliharaan terhadap informasi dan membuat informasi tersebut tersedia saat dibutuhkan. Basis data adalah suatu kumpulan dari data yang membentuk suatu berkas yang saling berkaitan dengan tata cara tertentu yang berfungsi untuk membentuk data baru atau informasi. Basis data disimpan dalam *hardware* komputer dengan *software* tertentu yang dimanipulasi untuk kegunaan tertentu.

II.2.6. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL, sebuah *database* mengandung satu

atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL adalah *Relational Database Management System (RDMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari *Postgre SQL* dan lima kali lebih cepat dibandingkan *Interbase* (Iman Saufik Suasana dan Jatu Suryantari, 2016 : 37).

II.2.7. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah “bahasa” yang sudah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti *C++*, *Java*, *C#*, atau *VB.NET*. Walaupun demikian, UML tetap dapat digunakan untuk modeling

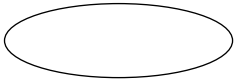
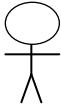
aplikasi prosedural dalam VB atau C. (Marbun Murni dan Sinaga Bosker, 2018, Hal : 35).


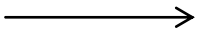
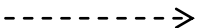
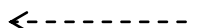
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>



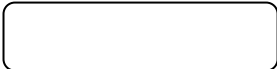
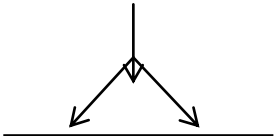
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

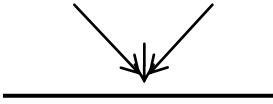
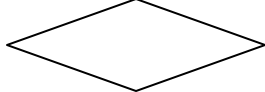
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.

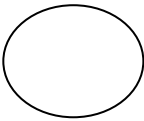
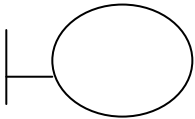
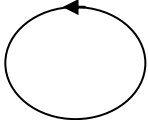

	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
<div style="border: 2px solid black; padding: 2px; display: inline-block;">New Swimline</div>	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

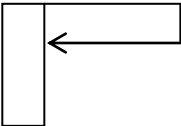

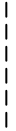
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .

	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 95)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang merupakan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Kelas (*Class*), *Relasi*, *Associations*, *Generalization dan Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 95)

II.2.8. Normalisasi

Menurut Indrajani (2015 : 11) Normalisasi adalah suatu teknik dengan pendekatan bottom-up yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu functional dependencies antara atribut. Tujuan utama normalisasi adalah mengidentifikasi kesesuaian hubungan yang mendukung data untuk memenuhi kebutuhan data perusahaan. Terdapat lima bentuk normal yang biasa digunakan yaitu :

1. *First Normal Form (1 NF)*

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal.

2. *Second Normal Form (2 NF)*

Untuk menjadikan tabel normal tingkatke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *fungsiional* tergantung pada semua atribut pembentuk *primary key*.

3. *Third Normal Form (3 NF)*

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung pada *primary key*.

4. *Fourth Normal Form (4 NF)*

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalued dependency*.

5. *Fifth Normal Form (5 NF)*

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika kata belter sebut dapat dipecah atau diproyeksikan menjadi beberapa table dan dari proyeksi-proyek siitu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula.