BAB II

LANDASAN TEORITIS

# **Teori terkait bahasan penilitian**

* + 1. **Sistem Penunjang Keputusan**

Menurut Heny Pratiwi (2016), Pengertian sistem pendukung keputusan yang dikemukan oleh McLeod (1998) yang menyatakan bahwa sistem pendukung keputusan merupakan sistem penghasil informasi yang ditujukan pada suatu masalah yang harus dibuat oleh manajer, sistem pendukung keputusan merupakan suatu sistem informasi yang ditujukan untuk membantu manajemen dalam memecahkan masalah yang dihadapinya. Defenisi selengkapnya adalah sistem penghasil informasi spesifik yang ditunjukan untuk memecahkan suatu masalah tertentu yang harus dipecahkan oleh manajer pada berbagai tingkatan. Sistem pendukung keputusan juga merupakan suatu sistem informasi berbasis computer yang menghasilkan berbagai alternative keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model. Sistem Pendukung Keputusan (*Decision Support System*) adalah sistem berbasis computer yang interaktif dalam membantu pengambilan keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur.

Menurut Kamalia Safitri, Dkk. (2017), Sistem pendukung keputusan adalah sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data yang digunakan untuk membantu pengambil keputusan pada situasi semi terstruktur dan tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat. Sistem pendukung keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi terstruktur yang spesifik. SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis *ad hoc* data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa.

# Tahapan Pengambilan Keputusan

Menurut Latifah, Dkk. (2017), Untuk menghasilkan keputusan yang baik ada beberapa tahapan proses yang harus dilalui dalam pengambilan keputusan terstruktur, semi-terstruktur, maupun tidak terstruktur, proses pengambilan keputusan melalui beberapa tahap berikut :

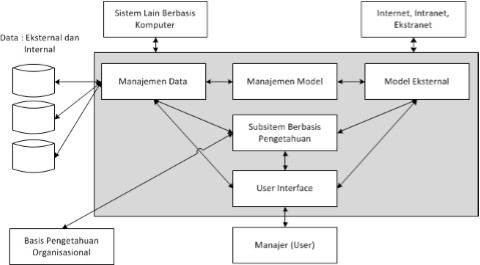
1. Tahapan Penelusuran Tahap ini pengambil keputusan mempelajari kenyataan yang terjadi, sehingga kita bisa mengidentifikasi masalah yang terjadi biasanya dilakukan analisis dari sistem ke subsistem pembentuknya sehingga didapatkan keluaran berupa dokumen pernyataan masalah.
2. Tahapan Desain Dalam tahap ini pengambil keputusan menemukan, mengembangkan dan menganalisis semua pemecahan yang mungkin yaitu melalui pembuatan model yang bisa mewakili kondisi nyata masalah. Dari tahapan ini didapatkan keluaran berupa dokumen alternatif solusi.
3. *Tahap Choice* Tahap ini pengambil keputusan memilih salah satu alternatif pemecahan yang dibuat pada tahap desain yang dipandang sebagai aksi yang paling tepat untuk mengatasi masalah yang sedang dihadapi. Dari tahap ini didapatkan dokumen solusi dan rencana implementasinya.
4. Tahap Implementasi Pengambil keputusan menjalankan rangkaian aksi pemecahan yang dipilih di tahap *choice*. Implementasi yang sukses ditandai dengan terjawabnya masalah yangdihadapi, sementara kegagalan ditandai masih adanya masalah yang sedang dicoba untuk diatasi. Dari tahap ini didapatkan laporan pelaksanaan solusi dan hasilnya.

# Komponen Sistem Pendukung Keputusan.

Menurut Latifah, Dkk. (2015), Sistem Pendukung Keputusan terdiri dari empat subsistem, yaitu:

1. Manajemen Data, meliputi basis data yang berisi data-data yang relevan dengan keadaan dan dikelola oleh perangkat lunak yang disebut dengan *Database Management System* (DBMS
2. Manajemen Model berupa sebuah paket perangkat lunak yang berisi model- model finansial, statistik, *management science*, atau model kuantitatif, yang menyediakan kemampuan analisa dan perangkat lunak manajemen yang sesuai.
3. Subsistem *Dialog* atau komunikasi, merupakan subsistem yang dipakai oleh user untuk berkomunikasi dan memberi perintah (menyediakan user interface).
4. Manajemen *Knowledge* yang mendukung subsistem lain atau berlaku sebagai komponen yang berdiri sendiri. Komponen ini dapat menyediakan keahlian yang diperlukan untuk memecahkan beberapa aspek masalah dan memberikan pengetahuan yang dapat meningkatkan operasi komponen sistem pendukung keputusan yang lain.

Berdasarkan semua definisi tersebut, sistem pendukung keputusan harus mencakup tiga komponen utama yaitu *DBMS (Database Management System*), manajemen model dan antarmuka pengguna. Subsistem manajemen *knowledge* adalah opsional, namun dapat memberikan banyak manfaat karena memberikan intelegensi bagi tiga komponen utama tersebut. Skematik komponen suatu system pendukung keputusan ditampilkan pada Gambar 2.1.



# Gambar 2.1. Komponen Sistem Pendukung Keputusan

*Sumber: (Latifah, dkk: 2015).*

# Karakteristik Sistem Pendukung Keputusan

Menurut Latifah Dkk. (2015), Peranan sistem pendukung keputusan dalam konteks kesuluruhan sistem informasi ditujukan untuk memperbaiki kinerja melalui aplikasi teknologi informasi. Berikut karakteristik yang diharapkan ada pada SPK:

1. Dukungan kepada pengambil keputusan, terutama pada situasi semi terstruktur dan tidak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk keputusan independen dan sekuensial. Keputusan bisa dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Dukungan di semua *fase* proses pengambilan keputusan: inteligensi, desain, pilihan, dan implementasi.
6. Dukungan di berbagai proses dan gaya pengambil keputusan.
7. Adaptivitas sepanjang waktu.
8. Pengguna merasa seperti di rumah.
9. Peningkatan efektivitas pengambilan keputusan (akurasi, timelines, kualitas) daripada peningkatan efisiensinya (biaya pengambilan keputusan).
10. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sendiri sistem sederhana.
12. Kapabilitas pemodelan memungkinkan eksperimen dengan berbagai strategi yang berbeda di bawah konfigurasi yang berbeda.

# Prosedur Pengambilan Keputusan

Menurut Slamet Hidayat Dkk. (2016), Ada beberapa prosedur pengambilan keputusan yaitu sebagai berikut :

1. Mengidentifikasi masalah.
2. Mengklarifikasi tujuan-tujuan khusus yang diinginkan.
3. Memeriksa berbagai kemungkinan untuk mencapai tujuan yang telah ditetapkan.
4. Mengakhiri proses itu dengan menetapkan pilihan bertindak dengan dasar fakta dan nilai (keputusan dan kemungkinan yang timbul.

# *Sales Counter*

Menurut Farrah Afsheena (2022), *Sales Counter* adalah salah satu profesi yang saat ini sedang diminati oleh banyak orang. Setiap perusahaan produksi, biasanya akan memiliki posisi pekerjaan ini untuk menjual produk di setiap konter secara lebih luas. Biasanya, seorang *sales counter* akan mempromosikan maupun menjual berbagai produk yang telah tersedia di konter sebuah toko, perusahaan, maupun dealer tertentu. Dengan adanya *sales counter* seperti ini, penjualan produk dari perusahaan diharapkan dapat terus mengalami peningkatan yang menguntungkan.

# Tugas *Sales Counter*

Menurut Farrah Afsheena (2022) Setiap profesi pasti akan memiliki tugas yang berbeda satu sama lain. Maka dari itu, ketika melamar pekerjaan, baik pihak perusahaan maupun pelamar wajib menyesuaikan posisi pekerjaan dengan kemampuan yang dimiliki oleh pelamar pekerjaan tersebut. Hal ini juga berlaku pada profesi *sales counter*. Profesi ini memiliki tugasnya tersendiri, yang akan sangat berkaitan dengan penjualan produk tertentu di sebuah konter. beberapa daftar tugas sales counter berikut ini:

1. Menjual Produk

Sesuai dengan namanya, salah satu tugas utama dari profesi *sales counter* adalah menjual produk-produk tertentu yang ada di konter yang telah disediakan oleh pihak perusahaan produksi. Biasanya, ada misi tertentu yang harus dipenuhi oleh seorang sales counter. Misal seperti setiap hari seorang sales counter harus bisa menjual produk dalam jumlah yang telah ditentukan oleh perusahaan. Namun, hal ini tidak akan berlaku di setiap perusahaan, karena setiap perusahaan pasti akan memiliki kebijakan yang berbeda-beda di dalamnya.

1. Menawarkan dan Memberi Informasi Lengkap terkait Produk yang Dijual.

Untuk meningkatkan angka penjualan produk, ada baiknya jika *sales counter* cermat dalam mencari cara untuk menawarkan produk yang sedang dijual tersebut. Penawaran ini wajib dilakukan dengan cara yang tepat, dan yang pasti harus bisa memikat hati para konsumen. Selain itu, dengan memberikan pelayanan terbaik kepada para konsumen pun juga bisa dianggap sebagai penawaran menarik yang dapat memikat hati para konsumen. Salah satu cara dalam memberikan pelayanan terbaik adalah dengan memberikan informasi lengkap terkait produk yang sedang dijual. Dalam hal ini, seorang *sales counter* dapat memberikan semua informasi secara rinci tentang produk tersebut, sehingga konsumen dapat menilai apakah produk tersebut dapat memenuhi kebutuhannya atau tidak.

1. Melaporkan Aktivitas Penjualan kepada Atasan

Aktivitas penjualan wajib dipantau secara tepat. Dalam hal ini, tidak boleh ada kesalahpahaman antara pihak *sales counter* maupun pihak atasan, sehingga disarankan bagi para *sales counter* untuk selalu melaporkan semua hasil dari aktivitas penjualan kepada atasan secara rutin. Laporan ini biasanya juga dapat berupa sebuah dokumen yang menampilkan semua kegiatan penjualan yang sudah terjadi. Di dalam dokumen ini nantinya akan terdapat semua data dari mulai jenis produk yang terjual, jumlah produk yang terjual, hingga nilai jual yang berhasil didapatkan. Semua data ini harus sesuai, sehingga tidak terjadi miss communication antara satu pihak dengan pihak lainnya.

1. Mempertahankan dan Menjaga Hubungan dengan Konsumen

Hubungan antara *sales counter* dengan konsumen bisa dibilang cukup erat. Sebab, konsumen akan melihat pelayanan yang diberikan oleh *sales counter* ketika mengunjungi sebuah konter tertentu. Dengan melihat pelayanannya ini, konsumen dapat menilai kualitas dari perusahaan tersebut. Dengan demikian, seorang *sales counter* wajib mempertahankan serta menjaga hubungannya dengan konsumen, supaya konsumen dapat kembali lagi untuk membeli produk yang ditawarkan lainnya. Tentu saja tugas yang satu ini akan membawa keuntungan yang maksimal terhadap perusahaan.

1. Mencari dan Mempersiapkan Prospek Baru

Tugas lain dari seorang *sales counter* ialah mencari dan mempersiapkan prospek baru para konsumen. Meskipun jika dilihat secara kasat mata tugas ini tergolong mudah, akan tetapi kenyataannya tugas ini cukup sulit untuk dilakukan.

1. Mengubah Pengunjung Menjadi Pembeli

Dari semua poin utama di atas, tugas yang paling vital bagi seorang *Sales Counter* adalah mengubah pengunjung yang datang menjadi pembeli dan bahkan menjadi pelanggan. Dalam beberapa lowongan yang ada, perusahaan menghendaki kandidat yang dapat melakukan hal ini.

1. **Metode Promethee**

Menurut Indriantoro & Utami, (2016) *PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations)* merupakan salah satu metode dari MCDM (*Multi Criteria Decision Making*) untuk penentuan urutan (prioritas) dalam analisis. Masalah pokoknya adalah kesederhanaan, kejelasan, dan kestabilan. Sangat tepat untuk digunakan karena dugaan dari dominasi kriteria yang digunakan dalam promethee adalah penggunaan nilai dalamhubungan *outrangking*. Sehingga diperoleh solusi atau hasil dari beberapa alternatif untuk diambil sebuah keputusan.

*Promethee* mempunyai kemampuan untuk menangani bayak perbandingan, pengambil keputusan hanya mendefinisikan skala ukurannya sendiri tanpa batasan, untuk mendefinisikan prioritasnya dan preferensi untuk setiap kriteria dengan memutuskan pada nilai (value), tanpa memikirkan tentang metode perhitungan.

Metode *promethee* menggunakan kriteria dan bobot dari masing- masing kriterianya yang kemudian diolah untuk menentukan pemilihan alternatif lapangan, yang hasilnya berurutan berdasarkan prioritasnya. Penggunaan metode *promethee* dapat dijadikan metode untuk pengambilan keputusan di bidang pemasaran, sumber daya manusia, pemilihan lokasi, atau bidang lain yang berhubungan dengan pemilihan alternatif. Prinsip yang digunakan adalah penetapan prioritas alternatif yang telah ditetapkan berdasarkan pertimbangan (∀*i* | f*i* (.) ℜ[Real]) dengan kaidah dasar:

Max {*f*1(x), *f*2(x), *f*3(x)…..*f*k(x) | x ∈ℜ}

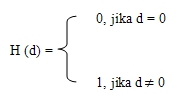
Di mana K adalah sejumlah kumpulan alternatif, *fi* = (*i* =1,2,3,4,5,…*.K* ) merupakan nilai/ukuran relatif kriteria untuk masing-masing alternatif. Dalam aplikasinya sejumlah kriteria telah diteteapkan unruk menjelaskan K yang merupakan penilaian dari ℜ(Real).

*Promethee* termasuk dalam keluarga dari metode outranking yang dikembangkan oleh B.Roy, dan meliputi dua fase:

* + - 1. Membangun hubungan outranking dari K
      2. Eksploitasi dari hubungan ini memberikan jawaban optimasi kriteria dalam paradigma permasalahan multikriteria.

Dalam metode *Promethee* disajikan enam bentuk fungsi preferensi kriteria. Hal ini tentu saja tidak mutlak, tetapi bentuk ini cukup baik untuk beberapa kasus, beberapa kriteria tersebut diantaranya:

1. Kriteria biasa (*Usual Criterion*)

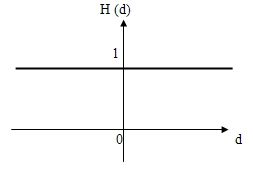
Pada referensi ini, tidak ada beda antara a dan b jika hanya jika f(a)=f(b), apabila nilai kriteria pada masing-masing alternatif memiliki nilai berbeda, pembuat keputusan membuat preferensi mutlak untuk alternatif memiliki nilai yang lebih baik.

Dimana d = selisih nilai kriteria {d = *f* (a) – *f* (b

Keterangan:

1. H(d): fungsi selisih kriteria alternatif
2. d: selisih nilai kriteria {d = f(a) – f(b)}

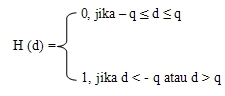
Pada kasus ini, tidak ada beda (sama penting) antara a dan b jika dan hanya jika *f* (a) = *f* (b); Fungsi H (d) untuk preferensi ini disajikan pada gambar 1.2:



**Gambar 2.2.** Kriteria Biasa

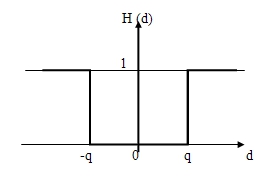
1. Kriteria *Quasi (Quasi Criterion)*

Dalam fungsi preferensi *quasi criterion* atau kriteria quasi, selisih hasil evaluasi untuk masing-masing nilai kriteria antar alternatif H (d) berpreferensi mutlak jika nilai H (d) dapat melebihi nilai q (Brans, 1982:181)



Keterangan:

1. H (d): fungsi selisih kriteria antar alternatif
2. D: selisih nilai kriteria {d = f(a) – f(b)}
3. Parameter (q): harus merupakan nilai tetap



**Gambar 2.3.** Kriteria *Quasi*

Gambar 1.3 menjelaskan dua alternatif memiliki preferensi yang sama penting selama selisih atau nilai H (d) dari masing-masing alternatif untuk kriteria tertentu tidak melebihi nilai q, dan apabila selisih hasil evaluasi untuk masing-masing alternatif melebihi nilai q maka terjadi bentuk preferensi mutlak.

1. Kriteria dengan Preferensi Linier

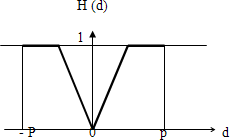
Kriteria preferensi linier dapat menjelaskan bahwa selama ini selisih memiliki nilai yang lebih rendah dari P, preferensi dari pembuat keputusan meningkat linier dengan nilai d (Brans, 1982:182)



Keterangan:

* 1. H (d): fungsi selisih kriteria antar alternatif
  2. d: selisih nilai kriteria {d = f(a) – f(b)}
  3. p: nilai kecenderungan atas.

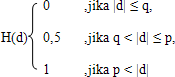
Jika nilai d lebih besar dibandingkan dengan nilai p, maka terjadi preferensi mutlak. Fungsi kriteria ini digambarkan pada gambar 1.4.

****

**Gambar 2.4.** Kriteria dengan Prefernsi Linier

1. Kriteria Level

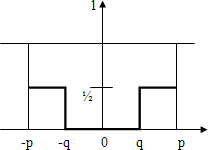
Dalam kasus ini kecenderungan tidak berbeda dengan q dan kecenderungan peferensi p ditentukan secara simultan. Jika d berbeda diantara nilai p dan q, hal ini berarti situasi preferensi yang lemah (H (d) = 0.5).

Fungsi ini disajikan pada:

Keterangan:

* 1. H (d): fungsi selisih kriteria antar alternatif
  2. P: nilai kecenderungan atas
  3. Parameter (q): harus merupakan nilai tetap

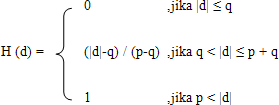
Gambar 1.5. menjelaskan pembuat keputusan telah menentukan kedua kecenderungan untuk kriteria ini.



**Gambar 2.5.** Kriteria Level

1. Kriteria dengan Preferensi Linier dan Area yang Tidak Berbeda

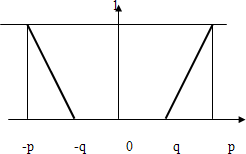
Pada kasus ini, pengambilan keputusan mempertimbangkan peningkatan preferensi secara linier dari tidak berbeda hingga preferensi mutlak dalam area antara dua kecenderungan q dan p (Brans, 1982:182).



Keterangan:

* + 1. H(d): fungsi selisih kriteria antar alternatif
    2. d: selisih nilai kriteria {d=f(a) – f(b)}
    3. Parameter (p): nilai kecenderungan atas
    4. Parameter (q): harus merupakan nilai yang tetap

Dua parameter p dan q telah ditentukan nilainya. Fungsi H adalah hasil perbandingan antar alternatif, seperti pada gambar 1.6:



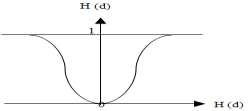
**Gambar 2.6.** Kriteria Preferensi Linier dan Area yang Tidak Berbeda

1. Kriteria Gaussian

Fungsi ini bersyarat apabila telah ditentukan nilai g, dimana dapat dibuat berdasarkan distribusi normal dalam statistik. Nilai H(d) tidak akan pernah bernilai satu (Brans, 1982:183).

H(d) = 1 – exp {-d2/2 2}

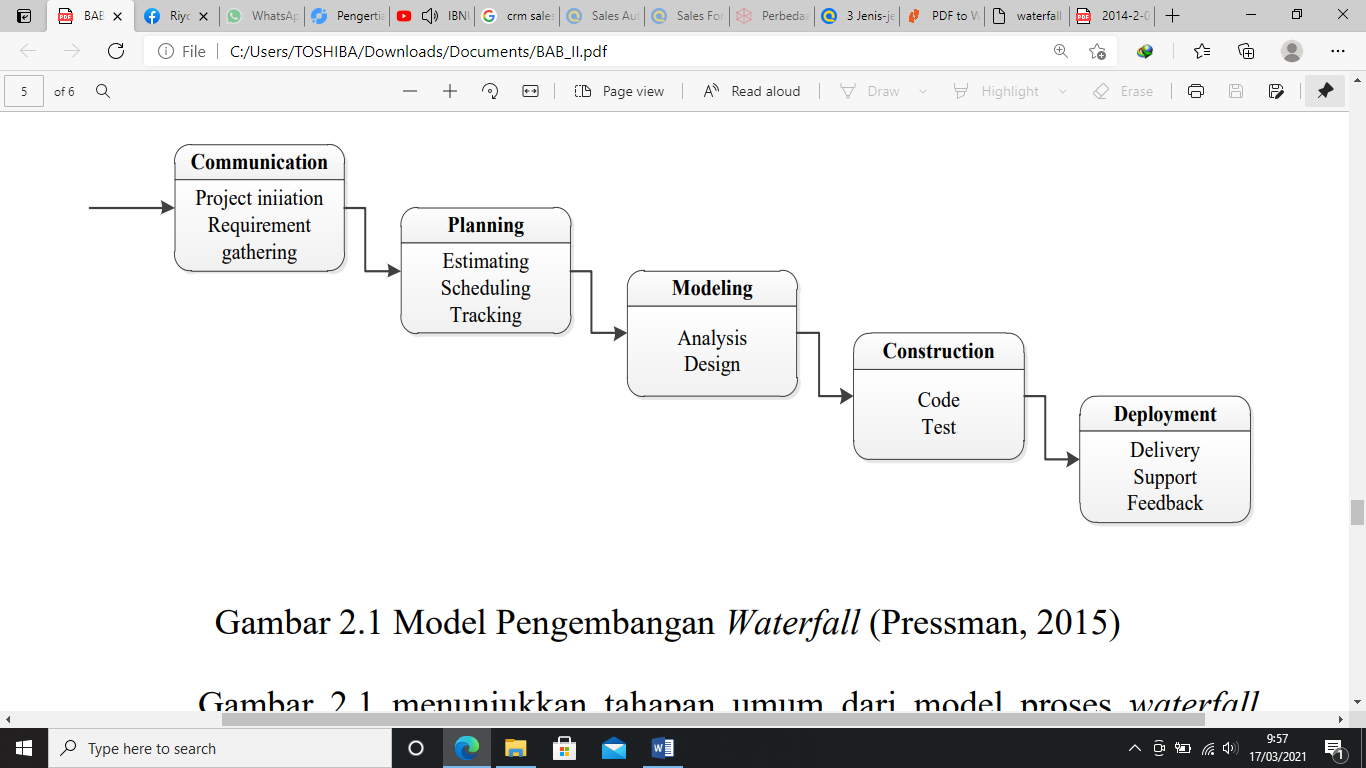
Pada penerapannya kriteria *Gaussian* akan digunakan pada distribusi normal statistik seperti penilaian terhadap tingkat keamanan lingkungan. Nilai g merupakan batas antara keamanan buruk sampai dengan tingkat aman sekali. Pada kriteria *Gaussian* tidak ada parameter yang tetap dalam menentukan nilai batas parameter (g).

****

**Gambar 2.7.** Kriteria *Gaussian*

# *Waterfall*

Proses-proses standar tersebut dituangkan dalam satu metode yang dikenal dengan nama *Waterfall* yang merupakan metodelogi umum dalam pengembangan sistem yang menandai kemajuan usaha dan desain. Menurut Pressman (2015:42), model *Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut juga dengan “classic life cycle” atau metode *Waterfall*. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam Software Engineering (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

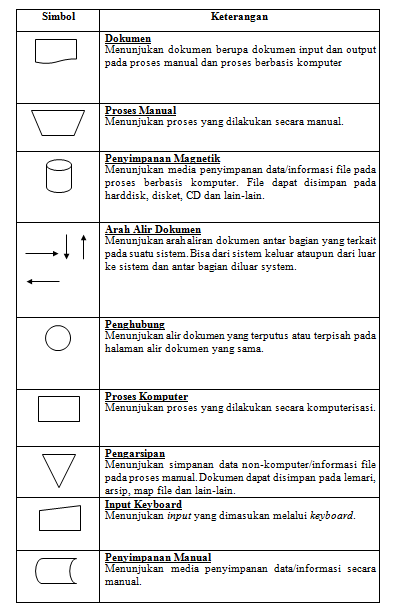


**Gambar 2.8.** Model Pengembangan *Waterfall* (Pressman, 2015)

1. ***Flowmap***

Menurut Meza Silvana, Dkk. (2015) *FlowMap* atau juga dapat disebut block chart atau *Flow Of Document* (FOD) /bagan alir merupakan penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Flowmap efektif digunakan untuk menggambarkan proses maupun prosedur dalam sebuah organisasi. Flowmap digunakan untuk menggambarkan urutan prosedur/proses kerja dalam pembuatan sistem.

Ada beberapa komponen *flowmap* yang harus di perhatikan dalam membangun sebuah *flowmap*, diantaranya bisa di lihat pada tabel 2.1 komponen *flowmap* :



**Tabel 2.1.** Komponen *flowmap*

1. ***Data Flow Diagram (DFD)***

Menurut Maniah dan Hamidin (2017:44) mengemukakan bahwa: “Diagram alir data (DFD), terutama untuk menggambarkan sistem operasi dimana fungsi sistem operasional dimana fungsi sistem sangat pending dan kompleks dibandingkan data yang dimanipulasi sistem.”

Sukamto dan Shalahuddin (2014:288), “Data Flow Diagram atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah refresentasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (input) dan keluaran (output). DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemograman berorientasi objek.”

Berikut adalah beberapa notasi simbol yang ada pada *Data Flow Diagram* :

|  |  |
| --- | --- |
| Notasi | Keterangan |
|  | Entitas luar (external entity) atau masukan (input) atau keluaran (output) atau orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan Catatan: Nama yang digunakan pada masukan (input) atau keluaran (output) biasanya berupa kata benda. |
|  | Aliran data; merupakan data yang dikirm antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (input) atau keluaran (output) Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa” |
|  | Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja |
|  | File atau basis data atau penyimpanan (storage); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data (Entity Relationship Diagram (ERD), Conceptual Data Model (CMD), Physical Data Model (PDM)) Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda. |

**TabelTabel 2.2.** Simbol-simbol *Data Flow Diagram* (Sumber: Maniah dan Hamidin (2017:44))

1. ***Entiti Relationship Diagram (ERD)***

Entity Relationship Diagram (ERD) merupakan sebuah cara untuk menggambarkan sebuah basis data yang menggunakan simbol-simbol beserta hubungan antara simbol-simbol tersebut.

Menurut Rosa dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”. Sedangkan Menurut Fatta dalam (Taufik, 2017) *Entity Relationship Diagram (ERD)* adalah gambar atau diagram yang menunjukan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis.”.

Dapat disimpulkan bahwa Entity Relationship Diagram (ERD) merupakan teknik penggambaran model jaringan suatu basis data dengan susunan data yang disimpan dalam sistem secara abstrak dengan menggunakan notasi dan simbol.

Simbol-simbol atau komponen-komponen yang digunakan dalam penggambaran Entity Relationship Diagram (ERD) yaitu:

1. Entitas (*Entity)*

Entity menunjukkan obyek-obyek dasar yang terkait didalam sistem obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data.

Nama\_Entitas

**Gambar 2.9.** Entitas (Sumber: Rosa dan Shalahuddin 2015:50)

1. Atribut (*Atribute)*

Atribut sering juga disebut sebagai properti, merupakan keteranganketerangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data

**Gambar 2.10.** Atribut (Sumber: Rosa dan Shalahuddin 2015:50)

1. Relasi (*Relation)*

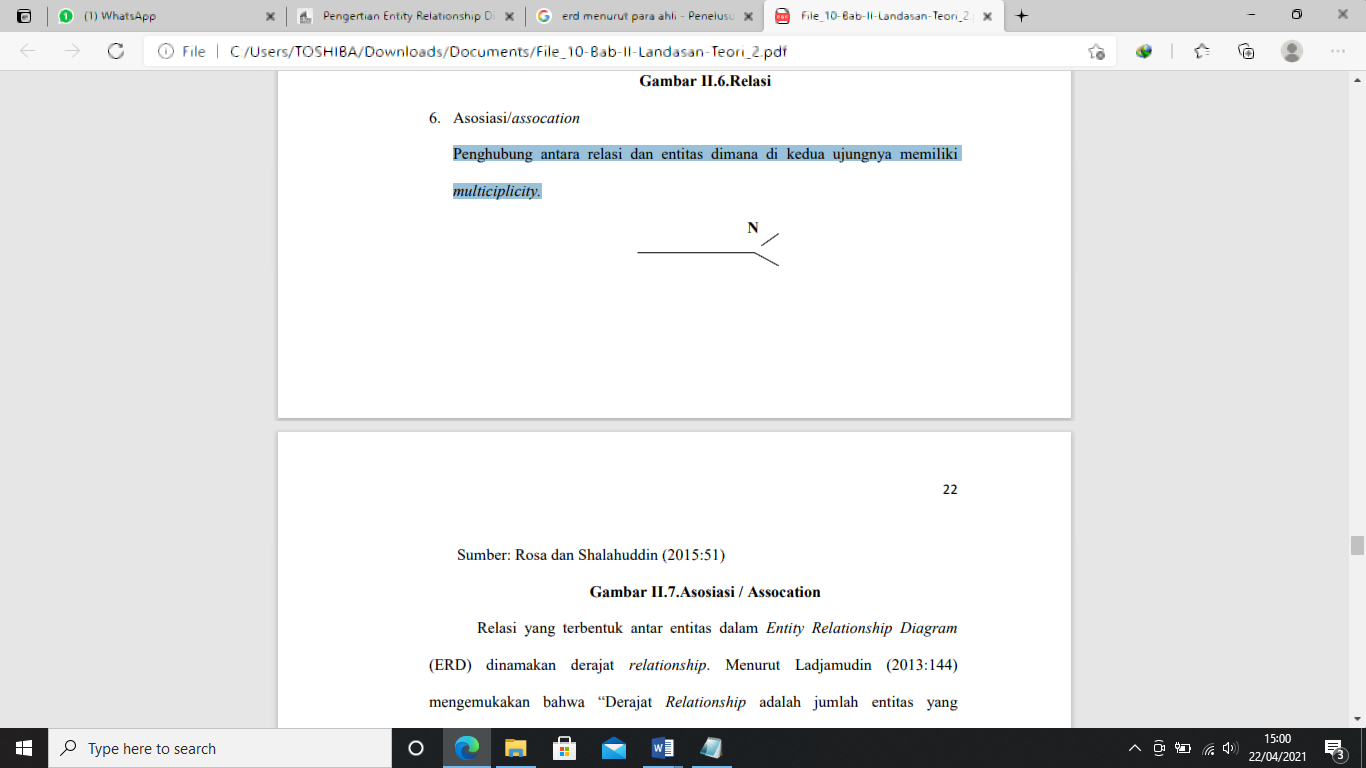
Relasi atau hubungan adalah kejadian atau transaksi yang terjadi diantara dua entity yang keterangannya perlu disimpan dalam basis data.

Nama Relasi

**Gambar 2.11.** Relasi (Sumber: Rosa dan Shalahuddin 2015:50)

1. Asosiasi (*Assocation)*

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiciplicity.



**Gambar 2.12.**  Relasi (Sumber: Rosa dan Shalahuddin 2015:50)

1. **Normalisasi**

Menurut Indrajani (2015:7), normalisasi adalah teknik dengan melakukan sebuah pendekatan bottom-up yang digunakan dalam membantu mengidentifikasikan hubungan. Sedangkan Menurut Lubis (2016:115) mengemukakan bahwa, “Normalisasi merupakan sebuah upaya untuk mendapatkan basis data dengan struktur yang baik, terutama sekali untuk efesiensi ruang penyimpanan (storage)”., adapun beberapa tahapan-tahapan dalam membuat normalisasi, diantaranya yaitu:

1. *Unnormalized Form* (UNF)

Adalah sebuah tabel yang memuat satu atau lebih kelompok yang berulang.

1. *First Normal Form* (1NF)

Adalah sebuah relasi yang terdiri dari perpotongan dari setiap baris dan kolom berisi satu dan hanya satu buah nilai saja. Aturan dari 1NF yaitu:

1. Tidak ada atribut multi-value, atribut komposit atau kombinasinya
2. Mendefinisikan atribut kunci c. Setiap atribut dalam table tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi)
3. *Second Normal Form* (2NF)

Adalah sebuah relasi yang berada dalam bentuk 1NF di mana setiap atribut yang bukan primary key bergantung secara fungsional penuh kepada primary key. Aturan dari 2NF yaitu :

1. Sudah memenuhi dalam bentuk normal kesatu (1NF)
2. Semua atribut bukan kunci hanya boleh tergantung (functional dependency) pada atribut kunci.
3. Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada table yang lain
4. Perlu ada table penghubung ataupun kehadiran foregin key bagi atribut-atribut yang telah dipisah tadi.
5. *Third Normal Form* (3NF)

Adalah relasi yang berada dalam bentuk 1NF dan 2NF di mana tidak ada lagi atribut yang bukan primary key yang bergantung secara transitif kepada primary key. Aturan dai 3NF yaitu:

1. Sudah berada dalam bentuk normal kedua (2NF).
2. Tidak ada ketergantungan transitif (di mana atribut bukan kunci tergantung pada atribut bukan kunci lainya)
3. ***Php***



**Gambar 2.13.**Logo *php* (Source : wikipedia)

Menurut Hidayatullah dan Kawistara dalam Fridayanthie dan Mahdiati (2016:130), “PHP merupakan script yang berintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*).PHP adalah script yang digunakan untuk membuat halaman web dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan”.

Sedangkan Menurut Holisatul Munawaroh dan Aeri Rachmad (2015:18), “PHP merupakan bahasa sever-side scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari server-side scripting adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan do server tetapi disertakan pada dokumen HTML”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, PHP adalah bahasa pemograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi *web* yang ditanamkan kedalam HTML, yang dijalankan di dalam sebuah *webserver*, dan juga bisa menghasilkan tampilan *website* yang dinamis.

1. ***Mysql***



**Gambar 2.14.**Logo *mysql* (Source : wikipedia)

Menurut Lubis, dkk. (2017:1537), “MySql adalah database yang unik untuk melakukan pendekatan yang berbeda untuk mentimpan dan mengakses data melalui konsep mesin penyimpann”.

Menurut Suprianto dalam Arif, dkk.(2017:1269), “MySql merupakan software yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *Open Source* yang menyatakan bahwa software ini dilengkapi dengan *source code*, selain tentu saja bentuk executable atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bias diperoleh dengan cara mendownload di internet secara gratis”.

Menurut Fauziah et all ( 2017:4), “MySql adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang database sebagai sumber dan pengelolaan datanya. Kepopuleran MySql antara lain karena MySql menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah digunakan. MySql juga bersifat open source kecuali pada *windows* yang bersifat *shareware*”.

1. ***Xampp***

**Gambar 2.15.** Logo Xampp (Source : wikipedia)

Menurut Purbadian (2016:1), berpendapat bahwa “XAMPP merupakan suatu software yang bersifat open source yang merupakan pengembangan dari LAMP (Linux, Apache, MySQL, PHP dan Perl)”.

Menurut Kartini (2013:27-26), “Xampp merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket”.

Sedangkan menurut (2015:1) “Xampp merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa Xampp merupakan tool pembantu pengembangan paket perangkat lunak berbasis open source yang menggabungkan Apache web server, MySQL, PHP dan beberpa modul lainnya di dalam satu paket aplikasi.

1. ***Black Box***

Menurut Rizky (dikutip oleh Hanifah, 2016:34), “Black Box Testing sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian black box ini menitik beratkan pada fungsi system”.

Berikut adalah 10 tipe pengujian dari metode Black Box menurut Julian (dikutip oleh Hanifah, 2016:34):

1. *Equivalence Partitioning*, Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.
2. *Boundary Value Analysis / Limit Testing*, Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari Equivalence Partitioning.
3. *Comparison Testing*, Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
4. *Sample Testing*, Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.
5. *Robustness Testing*, Data input dipilih diluar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
6. *Behavior Testing*, Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.
7. *Performance Testing*, Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi.
8. *Requirement Testing*, Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.
9. *Endurance Testing*, Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.
10. *Cause – Effect* Relationship Testing, Bagi-bagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja.
11. **White Box**

Menurut Rizky dikutip oleh Indriyani dalam jurnal PARADIGMA (2015:30-31). White box testing secara umum merupakan jenis testing yang lebih berkonsentrasi terhadap isi dari perangkat lunak itu sendiri. Jenis ini lebih banyak berkonsentrasi pada source code dari perangkat lunak yang dibuat sehingga membutuhkan proses testing yang jauh lebih lama dan lebih mahal dikarenakan membutuhkan ketelitian dari para tester serta kemampuan teknis pemrograman bagi para testernya. Berikut beberapa kelebihan White Box Testing:

1. Kesalahan logika Digunakan pada sintaks “if” dan pengulangan. Dimana white box testing akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.
2. Ketidaksesuaian asumsi Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk dianalisa dan diperbaiki.
3. Kesalahan ketik Mendeteksi bahasa pemrograman yang bersifat case sensitive. Kelemahan White Box Testing Untuk perangkat lunak yang tergolong besar, White Box Testing dianggap sebagai strategi yang tergolong boros karena akan melibatkan sumber daya yang besar untuk melakukannya.

Dengan pengujian white box testing dapat diketahui flow graph dan cyclomatix complexity. Untuk menghitungnya digunakan rumus:

V (G) = E – N + 2 atau V (G) = P + 1

Diketahui : E (jumlah edge pada flowgraph) = 4

N (jumlah node pada flowgraph) = 4

P (jumlah predikat node pada flowgraph) = 1

# Penelitian Sebelumnya

Penelitian terkait bertujuan untuk mengambil beberapa refrensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal lokal.

Berdasarkan hasil penelitian yang dilakukukan oleh Fathatul Fina K (2019) Dengan judul Implementasi Metode *Promethee* Pada Sistem Pendukung Keputusan Dalam Memberikan Reward Terhadarp Customer Pada Aplikasi Bengkel Studi Kasus : Dealer Honda Prima Motor Kuningan. Berdasarkan penelitian yang dilakukan, Dijelaskan bahwa telah dibangun sistem pendukung keputusan untuk memberikan reward kepada konsumen yang telah melakukan servis di bengkel honda Prima Motor Kuningan. Dari pengujian aplikasi yang dilakukan dengan mengimplementasikannya dapat membantu pihak pengambil keputusan untuk memutuskan konsumen mana yang berhak mendapatkan reward.

Lalu hasil penelitian yang pernah dilakukan oleh Nurul Azizah (2014) dengan judul Sistem Pendukung Keputusan Seleksi Calon Karyawan Dengan Metode *Promethee* Studi Kasus Pamella *Group* Yogyakarta. Berdasarkan penelitian yang dilakukan, dijelaskan bahwa telah dibangun sistem Penentuan Seleksi Calon Karyawan pada Pamella Group, untuk mengambil keputusan dalam melakukan seleksi calon karyawan. Aplikasi ini telah di uji coba kan kepada pengguna, dalam hal ini adalah pihak HRD pada Pamella Grup.

Selanjutnya penelitian yang dilakukan oleh Satriawaty Mallu (2015) dengan judul Sistem Pendukung Keputusan Penentuan Karyawan Kontrak Menjadi Karyawan Tetap Menggunakan Metode Topsis. Berdasarkan penelitian yang dilakukan, dijelaskan bahwa dari pengujian aplikasi yang dilakukan dengan mengimplementasikannya dapat membantu pihak pengambil keputusan untuk memutuskan apakah karyawan kontrak tersebut sudah memenuhi kriteria untuk menjadi karyawan tetap.

Dan yang terakhir penelitian yang dilakukan oleh Ayu Septiana Surya (2016) dengan judul Penerapan Metode *Promethee* Dalam Sistem Penunjang Keputusan Penentuan Penerima Beasiswa Bidik Misi Univeristas Halu Oleo Gorontalo. Berdasarkan penelitian yang di lakukan bahwa Metode *Promethee* dapat diterapkan dalam sistem pendukung keputusan seleksi penerima bantuan Bidik Misi, karena metode ini dapat memberikan rekomendasi prioritas penerima bantuan bidik misi sesuai dengan kriteria penilaian yang digunakan dalam bentuk perankingan. Perankingan dilakukan dengan tiga cara yaitu leaving flow, entering flow dan net flow. Setelah tiga cara tersebut ditentukan, maka hasil net flow yang akan menentukan urutan ranking dari setiap alternatif.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Penulis/ Komponen | Penulis | | | |
| Fathatul Fina K | Nurul Azizah | Satriawaty Mallu | Ayu Septiana Surya |
| Judul Penelitian | Implementasi Metode *Promethee* Pada Sistem Pendukung Keputusan Dalam Memberikan Reward Terhadarp Customer Pada Aplikasi Bengkel | Sistem Pendukung Keputusan Seleksi Calon Karyawan Dengan Metode *Promethee* | Sistem Pendukung Keputusan Penentuan Karyawan Kontrak Menjadi Karyawan Tetap Menggunakan Metode Topsis. | Penerapan Metode *Promethee* Dalam Sistem Penunjang Keputusan Penentuan Penerima Beasiswa Bidik Misi Univeristas Halu Oleo Gorontalo. |
| Lokasi Penelitian | Dealer Honda Prima Motor Kuningan. | Pamella *Group* Yogyakarta | PT. Gowa Motor *Group* | Univeristas Halu Oleo Gorontalo. |
| Metode Penelitian | *Promethee* | *Promethee* | Topsis | *Promethee* |

**Tabel 2.3.** Tabel perbandingan hasil penelitian

# Kerangka Teoritis

Kerangka teoritis adalah dukungan dasar teoritis sebagai dasar pemikiran dalam rangka pemecahan masalah yang dihadapi peneliti. Kerangka teoritis adalah bagian dari penelitian, tempat peneliti memberikan penjelasan tentang hal-hal yang berhubungan dengan variabel pokok, subvariabel, atau pokok masalah yang ada dalam penelitiannya.

Untuk kerangka teoritis dari sistem yang akan penulis buat bisa di liat pada Gambar 2.15 Kerangka teoritis di bawah ini.

1. Penilaian masih subjektif.
2. *Sales counter* kurang termotivasi.
3. Belum ada sistem pendukung.

**Identifikasi Masalah**

**Metode Pengembangan Sistem:**

Metode Waterfall dengan tahapan yang berurut yaitu:

1. Requirement Analysys
2. Design
3. Implementation
4. Testing
5. Maintenance

**Metode Pengumpulan Data :**

1. Wawancara
2. Observasi
3. Studi Kasus

**Metode Penyelesaian Masalah :**

Implementasi metode *Preference Ranking Organization Method for Enrichment Evalution* (PROMETHEE).

Promethee termasuk dalam keluarga dari metode outranking yang dikembangkan oleh B.Roy, dan meliputi dua fase:

1. Membangun hubungan outranking dari K
2. Eksploitasi dari hubungan ini memberikan jawaban optimasi kriteria dalam paradigma permasalahan multikriteria.

**Hasil yang diharapkan:**

1. Aplikasi dapat menampilkan sales counter terbaik dengan hasil dari perangkingan
2. Sistem terdapat dua tampilan antar muka berupa tampilan untuk admin dan pimpinan.
3. Sales lebih bersemangat dan termotivasi untuk hasil terbaik

**Solusi :**

Merancang sistem penunjang keputusan pemilihan *sales counter* terbaik dengan metode *promethee*

**Gambar 2.8**Kerangka Teoritis

**Gambar 2.16.** Kerangka Teoritis