# **BAB II**

# **TINJAUAN PUSTAKA**

## **Tinjauan Studi**

Klasifikasi menggunakan metode *K-Nearest Neighbor* merupakan bidang penelitian yang telah banyak dikembangkan saat ini. Berikut penelitian terkait yang menjadi referensi.

**Tabel 2.1.** Penelitian Tentang Klasifikasi menggunakan *K-Nearest Neighbor*

|  |  |  |
| --- | --- | --- |
| **Penulis/Tahun** | **Judul** | **Diskripsi Singkat** |
| Ardiyallah Akbar dkk, 2017. [9] | Klasifikasi Mutu Mutiara Berdasarkan Bentuk dan Ukuran menggunakan *K-Nearest Neighbor*. | Hasil akhir dari sistem ini adalah mampu menentukan kelas dan kualitas mutiara. Dari data sebanyak 25 yang terdiri dari 10 mutiara kualitas A, 10 mutiara kualitas AA, dan 5 mutiara kualitas AAA. Dengan menggunakan Metode K-NN (K-Nearest Neighbor) dan nilai K=1 mampu menghasilkan tingkat akurasi mencapai 92,30%. |
| Nobertus Krisandi dkk, 2015. [10] | Algoritma *K-Nearest Neighbor* dalam Klasifikasi Data Hasil Produksi Kelapa Sawit pada PT. Minimas Kecamatan Parindu. | Berdasarkan hasil penelitian data terklasifikasi dalam 6 cluster berdasarkan kemiripan hasil produksi dari 50 kelompok tani yang ada di KUD. HIMADO. Hasil produksi yang dominan adalah produksi dari kelompok tani kelapa sawit yang terletak pada C1 dengan 17 anggota dengan persentase 34% yaitu kelompok 1, 2, 33, 34, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,50 untuk nilai k=7. |
| Kevin Martha Rasepta, 2015. [11] | Klasifikasi Status Gizi Balita menggunakan Metode *K-Nearest Neighbor.* | Hasil model klasifikasi akan diimplementasikan pada sistem berbasis *web*. Hasilnya sistem yang dibangun sesuai harapan dan mampu membantu calon *user* mengklasifikasi status gizi balita dengan tingkat akurasi tertinggi sebesar 90% pada skenario 90:10 dan 80:20 dengan k=1 sampai k=3, rata-rata tingkat akurasi adalah 82,057%. |

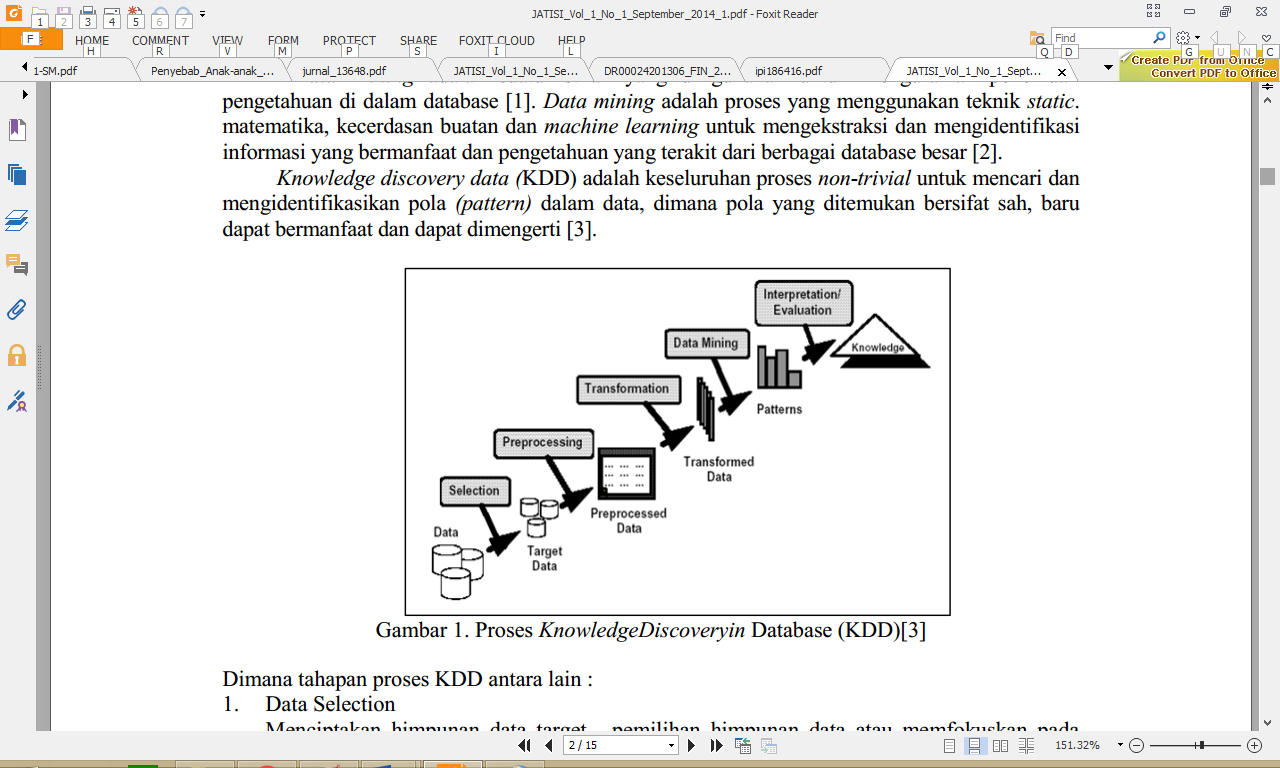
## **Tinjauan Teori**

### **Definisi Penyakit Masyarakat**

Perilaku penyimpangan adalah suatu tindakan yang dilakukan oleh seseorang atau sekelompok orang, yang tidak sesuai atau tidak menyesuaikan diri dengan norma-norma yang berlaku di masyarakat, baik yang dilakukan secara sadar ataupun tidak. Bentuk-bentuk penyimpangan tersebut, apabila terus berkembang akan menyebabkan timbulnya penyakit sosial dalam masyarakat. Dengan kata lain, penyakit sosial adalah bentuk penyimpangan terhadap norma masyarakat yang dilakukan secara terus-menerus [2].

### **Data Mining**

Menurut Han dan Kamber (2011), data mining adalah proses menemukan pola yang menarik dan pengetahuan dari data yang berjumlah besar. Menurut Linoff dan Berry (2011) Data mining adalah suatu pencarian dan analisa dari jumlah data yang sangat besar dan bertujuan untuk mencari arti dari pola dan aturan. Menurut Connolly dan Begg (2010), Data mining adalah suatu proses ekstraksi atau penggalian data yang belum diketahui sebelumnya, namun dapat dipahami dan berguna dari database yang besar serta digunakan untuk membuat suatu keputusan bisnis yang sangat penting. Dan menurut Vercellis (2009), Data mining adalah aktivitas yang menggambarkan sebuah proses analisis yang terjadi secara iteratif pada database yang besar, dengan tujuan mengekstrak informasi dan knowledge yang akurat dan berpotensial berguna untuk *knowledge wordkers* yang berhubungan dengan pengambilan keputusan dan pemecahan masalah. Istilah lain dari data (Han, 2006) yaitu *knowledge mining from database, knowladge extraction, data/patern analysis, data archeology, dan data dredging.* Banyak yang menggunakan data mining sebagai istilah populer dari KDD. *Knowledge discovery data* (KDD) adalah keseluruhan proses non-trivial untuk mencari dan mengidentifikasikan pola (*pattern*) dalam data, dimana pola yang ditemukan bersifat sah, baru dapat bermanfaat dan dapat dimengerti [12].



**Gambar 2.1:** Proses Knowledge Discoveryin Database (KDD)

(Sumber: Prasetyo, [12]).

Menurut Han dan Kamber [13], secara garis besar data mining dapat dikelompokkan menjadi 2 kategori utama, yaitu:

1. Predictive

*Predictive* merupakan proses untuk menemukan pola dari data dengan menggunakan beberapa variabel lain di masa depan. Salah satu teknik yang terdapat dalam *predictive* *mining* adalah klasifikasi. Tujuan dari tugas prediktif adalah untuk memprediksi nilai dari atribut tertentu berdasarkan pada nilai atribut-atribut lain. Atribut yang diprediksi umumnya dikenal sebagai target atau variable tak bebas, sedangkan atribut-atribut yang digunakan untuk membuat prediksi dikenal sebagai e*xplanatory* atau variable bebas. Contohnya, perusahaan retail dapat menggunakan data mining untuk memprediksikan penjualan dari produk mereka di masa depan dengan menggunakan data-data yang telah didapatkan dari beberapa minggu.

1. Descriptive

*Descriptive* dalam data mining merupakan proses untuk menemukan karakteristik penting dari data dalam suatu basis data. Tujuan dari tugas deskriptif adalah untuk menurunkan pola-pola (korelasi, *trend, cluster*, teritori, dan anomali) yang meringkas hubungan yang pokok dalam data. Tugas data mining deskriptif sering merupakan penyelidikan dan seringkali memerlukan teknik *post-processing* untuk validasi dan penjelasan hasil.

Menurut Hoffer, Ramesh & Topi [14], tujuan dari adanya data mining adalah:

1. *explanatory*, yaitu untuk menjelaskan beberapa kegiatan observasi atau suatu kondisi.
2. *confirmatory*, yaitu untuk mengkonfirmasi suatu hipotesis yang telah ada.
3. *exploratory*, yaitu untuk menganalisis data baru suatu relasi yang janggal.

Kegunaan data mining adalah untuk mengklasifikasikan pola yang harus ditemukan dalam data mining. Secara umum, data mining dapat diklasifikasikan dalam dua kategori yaitu deskriptif dan prediktif [15]. Adapun operasi-operasi dan teknik-teknik yang berhubungan:

1. Operasi *Predictive* modeling: (*classification, value prediction*)
2. *Database segmentation:* (*demographic clustering, neural clustering*)
3. *Link Analysis:* (*association discovery, sequential pattern discovery, similar time sequencediscovery*)
4. *Deviation detection:* (*statistics, visualization*)

Hasil dari data mining sering kali diintegrasikan dengan *decision support system* (DSS). Sebagai contoh, dalam aplikasi bisnis informasi yang dihasilkan oleh data mining dapat diintegrasikan dengan *tools* manajemen produk sehingga promosi pemasaran yang efektif yang dilaksanakan dan dapat diuji. Integrasi demikian memerlukan langkah *postprocessing* yang menjamin bahwa hanya hasil yang valid dan berguna yang akan digabungkan dengan DSS. Salah satu pekerjaan dan *postprocessing* adalah visualisasi yang memungkinkan analist untuk mengeksplor data dan hasil data mining dari berbagai sudur pandang. Ukuran-ukuran statistik dan metode pengujian hipotesis dapat digunakan selama *postprocessing* untuk membuang hasil data mining yang palsu. Gambar 2.2 menunjukkan hubungan data mining dengan area-area lain.



**Gambar 2.2:** Irisan Bidang Ilmu Data Mining

(Sumber: witten et al,,[12])

### **Proses Tahapan Data Mining**

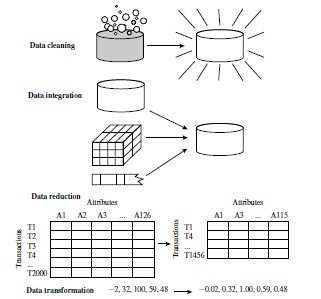
Menurut Han dan Kamber [13], Tahapan *Data Preprocessing* terbagi menjadi:

1. Data Preprocessing: An Overview

Pada bagian ini menyajikan gambaran dari *data preprocessing*. Pada bagian *data quality*, mengilustrasikan banyak unsur yang menentukan kualitas data. Ini memberikan insentif balik bagi *Data preprocessing* dan selanjutnya menguraikan tugas utama dalam *data preprocessing*

***Data Quality***: Data memiliki kualitas jika data tersebut memenuhi persyaratan dari penggunaan yang data yang dimaksudkan. Faktor-faktor yang terdiri dari kualitas data seperti akurasi, kelengkapan, konsistensi, ketepatan waktu, kepercayaan, dan *interpretability*. Banyak alasan yang memungkinkan untuk data yang tidak akurat (yaitu, memiliki nilai atribut yang salah). Kesalahan dalam transmisi data juga dapat terjadi. Kualitas data tergantung pada tujuan penggunaan data. Ketepatan waktu juga mempengaruhi kualitas data.

***Major Tasks in Data Preprocessing***: Langkah-langkah utama yang terlibat dalam *preprocessing* data, yaitu data pembersihan, integrasi data, reduksi data, dan transformasi data. Pembersihan data bekerja untuk "membersihkan" data dengan mengisi nilai-nilai yang hilang, *smoothing noisy* data, mengidentifikasi atau menghapus *outlier*, dan menyelesaikan inkonsistensi. Langkah *pre-processing* yang berguna adalah menjalankan data dengan pembersihan data. Berikut adalah Bentuk Data preprocessing



**Gambar 2.3:** Bentuk Data preprocessing

(Sumber: Han dan Kamber, [13]).

1. Data Cleaning

Pembersihan data (atau *data cleansing*) ber-upaya untuk mengisi nilai-nilai yang hilang, menghaluskan *noisy data*, mengidentifikasi *outlier*, dan inkonsistensi yang benar dalam data.

***Missing Values***: Banyak *tuple* yang tidak memiliki nilai yang tercatat ke dalam atribut. Cara mengatasi *missing values*:

1. Abaikan *tuple*: dilakukan ketika label kelas hilang. Metode ini sangat tidak efektif, kecuali *tuple* berisi beberapa atribut dengan nilai-nilai yang hilang. Dengan mengabaikan *tuple*, memungkinkan untuk tidak menggunakan nilai-nilai atribut yang tersisa dalam *tuple.*
2. Isikan nilai yang hilang secara manual: Secara umum, pendekatan ini memakan waktu dan mungkin tidak layak diberi *dataset* yang besar dengan banyak nilai-nilai yang hilang
3. Gunakan konstan global untuk mengisi nilai yang hilang: Ganti semua nilai atribut yang hilang dengan konstanta yang sama seperti label "*Unknown*".
4. Gunakan ukuran tendensi sentral untuk atribut (misalnya, rata-rata atau median) untuk mengisi nilai yang hilang.
5. Gunakan atribut berarti atau rata-rata untuk semua sampel milik kelas yang sama seperti *tuple* yang diberikan.
6. Gunakan nilai yang paling mungkin untuk mengisi nilai yang hilang: dapat ditentukan dengan regresi, alat berbasis inferensi menggunakan formalisme *Bayesian* atau *decision tree*.

***Noisy Data***: *Noise* adalah kesalahan acak atau varian dalam variabel yang diukur. Cara mengatasi *Noisy* Data:

1. *Binning*: pertama-tama melakukan pengurutan data dan partisi ke dalam (frekuensi yang sama) suatu tempat.
2. *Regression*: menghaluskan dengan mencocokkan data ke dalam fungsi regresi.
3. *Outlier Analysis*: Mendeteksi dan menghapus outlier.

***Data Cleaning as a Process***: Melakukan deteksi perbedaan data menggunakan metadata (domain, *range*, ketergantungan, distribusi), mendeteksi bagian *overloading*, mendeteksi *uniqueness rule*, *consecutive rule* dan *null*, menggunakan komersial *tools*. Data migrasi dan integrasi: memungkinkan transformasi yang ditentukan dengan data migrasi *tools* dan memungkinkan pengguna untuk menentukan transformasi melalui pengguna grafis dengan ETL *tools*. Integrasi dari dua proses: *Iterative* dan *Interactive*.

1. Data Integration

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau *file teks*. Integrasi data dilakukan pada atribut-aribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

1. Data Reduction

*Data Reduction* berguna untuk mendapatkan pengurangan representasi dari kumpulan data yang jauh lebih kecil di dalam volume tetapi belum menghasilkan hasil yang sama (atau hampir sama) dari suatu hasil analisis.

Teknik dalam *Data Reduction*:

1. Strategi *dimensionality reduction* pengurangan data meliputi *dimensionality reduction*, *numerosity reduction*, dan kompresi data.
2. *Wavelet Transform*: Data ditransformasikan ke jarak relatif antara obyek pada berbagai tingkat resolusi.
3. *Principal component Analysis*
4. *Attribute Subset Reduction*
5. *Regression* dan *Log linear models*
6. *Histogram*
7. *Clustering*
8. *Sampling*
9. *Data cube Agreggation*
10. Data Transformation and Data Discretization

Dalam *Data Transformation* dan *Data Discretization*, data diubah atau dikonsolidasikan sehingga proses *mining* yang dihasilkan mungkin lebih efisien, dan pola yang ditemukan mungkin lebih mudah untuk dipahami.

Strategi *Data Transformation*:

1. *Smoothing*, yang bekerja untuk menghilangkan *noise* dari dana data.
2. Atribut konstruksi (konstruksi atau fitur), di mana atribut baru dibangun dan ditambahkan oleh himpunan atribut untuk membantu proses *mining*.
3. Agregasi, dimana ringkasan atau agregasi operasi diterapkan pada data.
4. Normalisasi, dimana data atribut adalah skala sehingga jatuh dalam kisaran yang lebih kecil.
5. *Discretization,* dimana nilai-nilai baku dari atribut numerik (misalnya, usia) akan diganti dengan label Interval (misalnya, 010, 11-20, dll) atau label konseptual (misalnya, remaja, dewasa, senior).
6. Generasi hirarki konsep untuk data nominal, di mana atribut dapat digeneralisasi untuk konsep-tingkat yang lebih tinggi, seperti kota atau negara.

### **Teknik Data Mining**

Teknik data mining terbagi menjadi tiga, yaitu*: Association Rule Mining, Classification, Clustering* dan *Regretion.*

1. Association Rule Mining

Menurut Olson dan Shi [16], *Association Rule Mining* merupakan teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi item atau untuk menemukan hubungan hal tertentu dalam suatu transaksi data dengan hal lain di dalam transaksi, yang digunakan untuk memprediksi pola. Sedangkan menurut Han dan Kamber [13], *Association Rule Mining* terdiri dari itemset yang sering muncul. *Association Rule Mining* dapat dianalisa lebih lanjut untuk mengungkap aturan korelasi untuk menyampaikan korelasi statistik antara *itemsets* A dan B.

1. Classification

Menurut Olson dan Shi[16],Klasifikasi *(Classification),* metode-metodenya ditunjukan untuk pembelajaran fungsi-fungsi berbeda yang memetakan masing-masing data terpilih ke dalam salah satu dari kelompok kelas yang telah ditetapkan sebelumya. Menurut Han dan Kamber [13], *Classification* adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui.

Dasar pengukuran untuk mengukur kualitas dari penemuan teks, yaitu:

1. *Precision*: tingkat ketepatan hasil klasifikasi terhadap suatu kejadian.
2. *Recall*: tingkat keberhasilan mengenali suatu kejadian dari seluruh kejadian yang seharusnya dikenali.
3. *F-Measure* adalah nilai yang didapatkan dari pengukuran *precision* dan *recall* antara *class* hasil *cluster* dengan *class* sebenarnya yang terdapat pada data masukan.
4. Clustering

Menurut Han dan Kamber [13], *Clustering* adalah proses pengelompokkan kumpulan data menjadi beberapa kelompok sehingga objek di dalam satu kelompok memiliki banyak kesamaan dan memiliki banyak perbedaan dengan objek dikelompok lain. Perbedaan dan persamaannya biasanya berdasarkan nilai atribut dari objek tersebut dan dapat juga berupa perhitungan jarak. *Clustering* sendiri juga disebut *Unsupervised* *Classification*, karena *clustering* lebih bersifat untuk dipelajari dan diperhatikan. *Cluster analysis* merupakan proses partisi satu set objek data ke dalam himpunan bagian. Setiap himpunan bagian adalah *cluster*, sehingga objek yang di dalam *cluster* mirip satu sama dengan yang lainnya, dan mempunyai perbedaan dengan objek dari *cluster* yang lain. Partisi tidak dilakukan dengan manual tetapi dengan algoritma *clustering*. Oleh karena itu, *Clustering* sangat berguna dan bisa menemukan *group* yang tidak dikenal dalam data.

Teknik *clustering* umumnya berguna untuk merepresentasikan data secara visual, karena data dikelompokkan berdasarkan kriteria-kriteria umum. Dari representasi target tersebut, dapat dilihat adanya kecenderungan lebih tingginya jumlah lubang pada bagian-bagian atau kelompok-kelompok teretentu dari target tersebut.

1. Regresi

Menurut Han dan Kamber [13]. Regresi merupakan fungsi pembelajaran yang memetakan sebuah unsur data ke sebuah variabel prediksi bernilai nyata.

### **Klasifikasi**

Klasifikasi merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri bisa berupa aturan “jika-maka”, berupa *decision tree*, *formula matematis* atau *neural network*. Metode-metode klasifikasi antara lain C4.5, *RainForest*, *Naïve Bayesian*, *neural network*, *genetic algorithm*, *fuzzy*, *case-based reasoning*, dan k-*Nearest Neighbor* [5], [6].

### **Metode *K-Nearest Neighbor* (K-NN)**

Algoritma k-NN adalah suatu metode yang menggunakan algoritma *supervised* [6], [8], [13], [17], [18]. Perbedaan antara *supervised learning* dengan *unsupervised learning* adalah pada *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data [6], [8], [13], [17]. Tujuan dari algoritma *k*-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *training samples* [8], [13]. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada *k*-NN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma *k*-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru [6], [8], [13]. Jarak yang digunakan adalah jarak *Euclidean Distance*. Jarak *Euclidean* adalah jarak yang paling umum digunakan pada data numerik [7]. *Euclidean distance* didefinisikan sebagai berikut [13]:

(1)

Keterangan:

: Jarak *Euclidean* (*Eucledean Distance*)

: *Record* ke-*i*

: *Record* ke-*j*

: data ke-*r*

:

Algoritma *k*-NN adalah algoritma yang menentukan nilai jarak pada pengujian *data testing* dengan *data training* berdasarkan nilai terkecil dari nilai ketetanggaan terdekat [7] didefinisikan sebagi berikut:

(2)

### **Penerapan Metode K-NN**

Penelitian yang dilakukan oleh Nobertus Krisandi dkk, 2015. [10]. Judul penelitian Algoritma *K-Nearest Neighbor* dalam Klasifikasi Data Hasil Produksi Kelapa Sawit pada PT. Minimas Kecamatan Parindu. Berikut adalah data hasil produksi 50 kelompok tani kelapa sawit pada PT. MINAMAS periode  
Juli-Desember 2011 yang terletak di Kecamatan Parindu Kabupaten Sanggau:

**Tabel 2.2.** Data Hasil Produksi Pada KUD. HIMADO Periode

Juli 2011-Desember 2011

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kel. | Jumlah Produksi (*Tonase*) | | | | | | Kel. | Jumlah Produksi (*Tonase*) | | | | | |
| Juli | Ags | Sep | Okt | Nov | Des | Juli | Ags | Sep | Okt | Nov | Des |
| 1. | 23 | 22 | 23 | 26 | 24 | 22 | 26. | 25 | 19 | 23 | 25 | 25 | 30 |
| 2. | 17 | 23 | 25 | 31 | 45 | 36 | 27. | 21 | 21 | 25 | 18 | 24 | 23 |
| 3. | 19 | 20 | 19 | 23 | 25 | 20 | 28. | 24 | 27 | 29 | 31 | 29 | 31 |
| 4. | 23 | 33 | 36 | 31 | 38 | 30 | 29. | 15 | 24 | 14 | 14 | 22 | 23 |
| 5. | 25 | 34 | 28 | 40 | 35 | 25 | 30. | 14 | 16 | 21 | 15 | 30 | 24 |
| 6. | 20 | 22 | 21 | 26 | 30 | 27 | 31. | 27 | 25 | 18 | 19 | 20 | 17 |
| 7. | 21 | 26 | 30 | 39 | 38 | 36 | 32. | 10 | 11 | 10 | 11 | 16 | 14 |
| 8. | 18 | 15 | 16 | 22 | 23 | 20 | 33. | 15 | 15 | 18 | 24 | 23 | 17 |
| 9. | 28 | 28 | 27 | 34 | 52 | 46 | 34. | 15 | 16 | 19 | 21 | 22 | 15 |
| 10. | 18 | 24 | 25 | 31 | 32 | 30 | 35. | 16 | 18 | 17 | 23 | 24 | 29 |
| 11. | 21 | 27 | 30 | 29 | 26 | 18 | 36. | 26 | 24 | 25 | 32 | 37 | 36 |
| 12. | 12 | 17 | 15 | 17 | 25 | 19 | 37. | 34 | 36 | 36 | 30 | 39 | 32 |
| 13. | 14 | 16 | 16 | 20 | 27 | 21 | 38. | 16 | 24 | 28 | 27 | 28 | 26 |
| 14. | 22 | 31 | 30 | 32 | 37 | 24 | 39. | 11 | 12 | 11 | 14 | 15 | 10 |
| 15. | 57 | 57 | 58 | 84 | 96 | 64 | 40. | 19 | 20 | 15 | 19 | 22 | 19 |
| 16 | 31 | 32 | 33 | 41 | 39 | 41 | 41. | 15 | 11 | 13 | 13 | 17 | 13 |
| 17. | 24 | 16 | 18 | 17 | 15 | 17 | 42. | 15 | 22 | 25 | 16 | 28 | 21 |
| 18. | 47 | 45 | 54 | 52 | 48 | 51 | 43. | 9 | 11 | 12 | 15 | 16 | 11 |
| 19. | 31 | 33 | 29 | 29 | 28 | 25 | 44. | 18 | 16 | 13 | 12 | 13 | 10 |
| Kel. | Jumlah Produksi (*Tonase*) | | | | | | Kel. | Jumlah Produksi (*Tonase*) | | | | | |
| Juli | Ags | Sep | Okt | Nov | Des | Juli | Ags | Sep | Okt | Nov | Des |
| 20. | 30 | 28 | 37 | 41 | 47 | 34 | 45. | 21 | 22 | 19 | 20 | 23 | 20 |
| 21. | 31 | 35 | 34 | 45 | 40 | 34 | 46. | 7 | 8 | 9 | 10 | 13 | 9 |
| 22. | 29 | 28 | 27 | 30 | 31 | 34 | 47. | 29 | 25 | 17 | 19 | 22 | 21 |
| 23. | 23 | 23 | 20 | 27 | 22 | 25 | 48. | 14 | 15 | 17 | 20 | 22 | 16 |
| 24. | 26 | 29 | 32 | 39 | 41 | 36 | 49. | 9 | 9 | 10 | 13 | 15 | 12 |
| 25. | 31 | 24 | 34 | 24 | 46 | 44 | 50. | 8 | 7 | 8 | 9 | 11 | 7 |

(Sumber: KUD. HIMADO Kec. Parindu)

Data hasil produksi dari kelompok tani yang berjumlah 50 *record* akan dibagi ke dalam *data training* dan *data testing*. Pembagian dilakukan dengan persentase, yaitu 70 % pada *data training* dan 30% pada *data testing* [4]. Nilai persentase tersebut juga dapat disesuaikan dengan *record* data. Setelah dilakukan proses pembagian data tersebut maka diperoleh 36 *record* data pada *data training*  
dan 14 *record* data pada *data testing.* Berdasarkan pembagian tersebut maka nilai *r*1*, r*2, *r*3,..., r36 terletak pada *data training* dan *r*37, *r*38*,..., r*50 terletak pada *data testing*. Langkah selanjutnya adalah dengan menghitung jarak *Euclidean* pada *data training* dengan menggunakan rumus *Euclidean Distance* dengan nilai *k*=1. Hasil perhitungan ini berpengaruh pada penempatan setiap data pada *cluster-cluster* yang di bentuk. Seluruh hasil perhitungan jarak *Euclidean* dengan *k*=1 ditunjukkan pada Tabel 2.3. Setelah nilai jarak *Euclidean* pada *data training* diketahui, maka dapat ditentukan *cluster-cluster* pada *data training* berdasarkan kedekatan nilai-nilai yang dihasilkan dari perhitungan jarak *Euclidean* tersebut. Banyaknya jumlah *cluster* ditentukan berdasarkan rumus H. A. Sturges. Dengan  
menggunakan rumus H. A. Sturges maka *cluster* yang di bentuk adalah sebanyak 6 *cluster*.

Adapun cluster-cluster tersebut adalah sebagai berikut:

= nilai dengan rentang jarak *Euclidean* d < 13,5

= nilai dengan rentang jarak *Euclidean* 13,5 ≤ d < 27

= nilai dengan rentang jarak *Euclidean* 27,0 ≤ d < 40,5

= nilai dengan rentang jarak *Euclidean* 40,5 ≤ d < 54

= nilai dengan rentang jarak *Euclidean* 54,0 ≤ d < 67,5

= nilai dengan rentang jarak *Euclidean* 67,5 ≤ d < 81

Peletakkan anggota *cluster* didasarkan pada besaran nilai jarak *Euclidean* pada setiap *record pada data training*. Penentuan *Cluster-cluster* mengacu pada nilai rentang dari *cluster-cluster* tersebut. Apabila nilai jarak *Euclidean* suatu *record* berada di dalam salah satu rentang yang telah diketahui, maka *record* tersebut akan diletakkan pada *cluster* yang bersangkutan. Berdasarkan hasil perhitungan maka *cluster-cluster* pada *data training* ditunjukkan pada Tabel 2.3.

**Tabel 2.3**. *Cluster Data Training*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Record*** | **Nilai** | ***Cluster*** | ***Record*** | **Nilai** | ***Cluster*** |
| *d*(1,2) | 15,36 | C1 | *d*(19,20) | 26,00 | C2 |
| *d*(3,4) | 28,40 | C3 | *d*(21,22) | 20,19 | C2 |
| *d*(5,6) | 21,04 | C2 | *d*(23,24) | 28,37 | C3 |
| *d*(7,8) | 33,11 | C3 | *d*(25,26) | 31,93 | C3 |
| *d*(9,10) | 28,01 | C3 | *d*(27,28) | 18,62 | C2 |
| *d*(11,12) | 23,49 | C2 | *d*(29,30) | 14,79 | C2 |
| *d*(13,14) | 27,16 | C3 | *d*(31,32) | 25,25 | C2 |
| *d*(15,16) | 86,92 | C6 | *d*(33,34) | 5,83 | C1 |
| *d*(17,18) | 79,89 | C6 | *d*(35,36) | 27,18 | C3 |

Setelah penempatan *cluster-cluster* pada *data training* selesai, selanjutnya dilakukan pengujian data dengan menggunakan *data testing.* Nilai *k* yang digunakan adalah *k*=3, *k*=5 dan *k*=7. Kemudian dihitung jarak *data testing* yang terdekat dengan *data training* dengan menggunakan rumus *Euclidean distance*. Untuk proses penentuan letak *cluster* dari *data testing* tersebut*,* ditentukan menurut kedekatan nilai *jarak euclideannya*. Hal ini diperoleh berdasarkan nilai minimum diantara nilai hasil pengujian dari *data-data testing* tersebut [19]. Dari hasil perhitungan *data testing* dengan *k*=3, *k*=5 dan *k*=7 diperoleh letak kelas pada *data testing* yang ditunjukkan pada Tabel 2.4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Record*** | **Nilai** | | | **Cluster** | | |
| ***k* = 3** | ***k* = 5** | ***k* = 7** | ***k* = 3** | ***k* = 5** | ***k* = 7** |
|  | 16,52 | 16,52 | 16,52 |  |  |  |
|  | 15,43 | 13,30 | 13,30 |  |  |  |
|  | 32,97 | 15,46 | 12,21 |  |  |  |
|  | 13,53 | 6,08 | 6,08 |  |  |  |
|  | 5,92 | 5,92 | 5,92 |  |  |  |
|  | 13,19 | 12,65 | 12,45 |  |  |  |
|  | 6,78 | 3 | 3 |  |  |  |
|  | 7,75 | 7,75 | 7,75 |  |  |  |
|  | 13,93 | 5 | 5 |  |  |  |
|  | 7,75 | 7,75 | 7,75 |  |  |  |
|  | 8,94 | 8,94 | 8,94 |  |  |  |
|  | 10,91 | 12,33 | 10,77 |  |  |  |
|  | 5,29 | 5,29 | 3,74 |  |  |  |
|  | 8,12 | 3,46 | 3,46 |  |  |  |

Menurut Tabel 2.4, dapat dilihat adanya perubahan pada keanggotaan *cluster-cluster*. Pada *k*=5*, r*38 dan *r*45 berpindah dari C2 ke C1. Untuk *r*39 berpindah dari C3 ke C2. Sedangkan pada *k*=7, *r*39 berpindah dari C2 ke C1. Hasil klasifikasi yang digunakan adalah hasil klasifikasi dengan nilai *k*=7. Hal ini dikarenakan kecendurangan *data testing* yang bergerak pada *cluster* dengan nilai *Euclidean* distance terendah yaitu C1. Dari seluruh hasil perhitungan pada *data training* dan *data testing*, maka seluruh *record* telah terklasifikasi ke dalam *cluster*nya masing-masing. Hasil rekapitulasi *cluster* data dan keanggotaannya ditunjukkan pada Tabel 2.5.

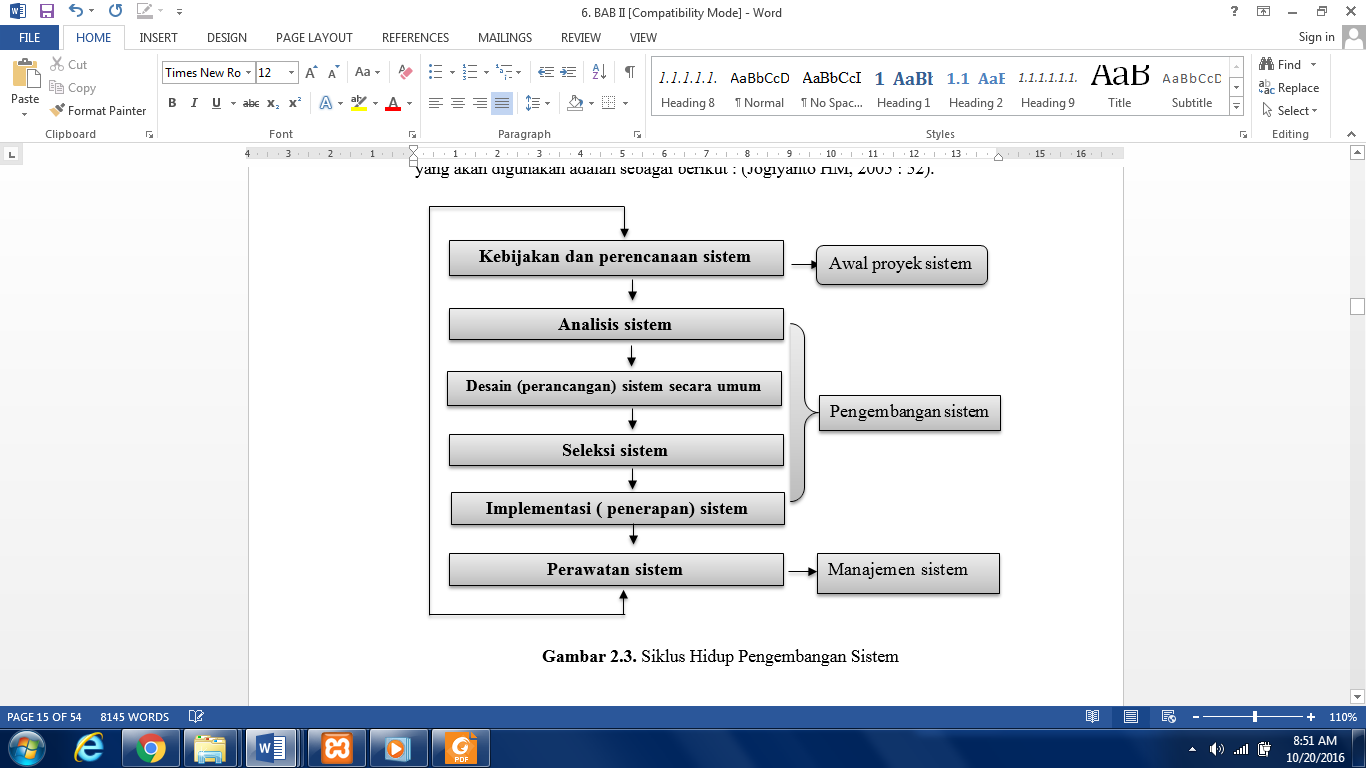
**Tabel 2.5**. Rekapitulasi Hasil *Cluster*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***k* = 3** | | ***k* = 5** | | ***k* = 7** | |
| ***Cluster*** | **Anggota** | ***Cluster*** | **Anggota** | ***Cluster*** | **Anggota** |
|  | 1,2,33,34,41,42,43,  44,46,47,48,49,50 |  | 1,2,33,34,38,40,  41,42,43,44,45,  46,47,48,49,50 |  | 1,2,33,34,38,39,40, 41,42,43,44,45,46, 47,48,49,50 |
|  | 5,6,11,12,19,20,21,  22,27,28,29,30,31,  32,37,38,40,45 |  | 5,6,11,12,19,20, 21,22,,27,28,29, 30,31,32,37,39 |  | 5,6,11,12,19,20,21, 22,,27,28,29,30,31, 32,37 |
|  | 3,4,7,8,9,10,13,14,  23,24,25,26,35,36,39 |  | 3,4,7,8,9,10,13, 14,23,24,25,26, 35,36, |  | 3,4,7,8,9,10,13,14, 23,24,25,26,35,36 |
|  | - |  | - |  | - |
|  | - |  | - |  | - |
|  | 15,16,17,18 |  | 15,16,17,18 |  | 15,16,17,18 |

Menurut Tabel 2.5, hasil klasifikasi untuk *k*=3 pada *cluster* pertama (C1) beranggotakan 13 *record* kelompok tani dengan persentase keanggotaan 26%. Pada *cluster* kedua (C2) beranggotakan 18 *record* kelompok tani dengan persentase keanggotaan 36%. Pada *cluster* ketiga (C3) beranggotakan 15 *record*  
kelompok tani dengan persentase keanggotaan 30%. Sedangkan pada *cluster* keempat (C4) dan kelima (C5) tidak memiliki anggota *cluster*, maka persentase keanggotaannya 0%. Sedangkan pada *cluster* keenam (C6) beranggotakan 4 *record* kelompok tani dengan persentase keanggotaan 8%. Hasil klasifikasi untuk *k*=5 pada *cluster* pertama (C1) beranggotakan 16 *record* kelompok tani dengan  
persentase keanggotaan 32%. Pada *cluster* kedua (C2) beranggotakan 16 *record* kelompok tani dengan persentase 32%. Pada *cluster* ketiga (C3) beranggotakan 14 *record* kelompok tani dengan persentase keanggotaan 28%. Sedangkan pada *cluster* keempat (C4) dan kelima (C5) tidak memiliki anggota *cluster*, maka persentase keanggotaannya adalah 0%. Sedangkan pada *cluster* keenam (C6)  
beranggotaan 4 *record* kelompok tani dengan persentase keanggotaan 8%. Hasil klasifikasi untuk *k*=7 pada *cluster* pertama (C1) beranggotakan 17 *record* kelompok tani dengan persentase keanggotaan 34%. Pada *cluster* kedua (C2) beranggotakan 15 *record* kelompok tani dengan persentase keanggotaan 30%. Pada *cluster* ketiga (C3) beranggotakan 14 *record* kelompok tani dengan persentase keanggotaan 28%. Sedangkan pada *cluster* keempat (C4) dan kelima (C5) tidak memiliki anggota, maka persentase keanggotaannya 0%. Pada *cluster* keenam beranggotakan 4 *record* kelompok tani dengan persentase keanggotaan 8%.

### **Siklus Hidup Pengembangan Sistem**

Menurut Sutabri Tata [20], suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah pada tahapan tersebut dalam proses pengembangan sistem.



**Gambar 2.5:** Siklus pengembangan hidup

(Sumber: Sutabri Tata. [20])

### **Analisis Sistem**

Analisa sistem (*System Analisa*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Analisa sistem adalah spesialis yang mempelanjari masalah dan kebutuhan sebuah organisasi untuk menentukan bagaimana orang, data, proses dan teknologi informasi dapat mencapai kemajuan terbaik untuk bisnis.

Analisis sistem adalah *Stakeholder* yang berperan sebagai fasilitator atau pelatih, menjebatani jurang komunikasi yang dapat secara alamiah berkembang antara pemilik dan pengguna *system nonteknis* atau desainer dan perkembangan sistem teknis.

Whitten, et al. [21] mengungkapkan “*System analysis* adalah study domain masalah bisnis untuk merekomendasikan perbaikan dan menspesifikasi persyaratan dan prioritas bisnis untuk solusi”.

Impak teknologi objek sangat berarti dalam dunia analisis dan desain sistem. Sebelum ada teknologi objek, kebanyakan bahasa pemrograman didasarkan pada apa yang disebut metode yang terstruktur (*structured method*). Contohnya COBOL bahasa yang domain 0, C, Fortan, Pascal, dan PL/i. Maka, metode analisis dan desain berorientasi objek telah muncul sebagai pendekatan terpilih untuk membangun kebanyakan sistem informasi saat ini.

Sebagai tambahan keahlian analisis dan desain sistem formal, seorang analis harus mengembangkan atau memilki keahlian lain, pengetahuan, dan karakter untuk menyelesaikan pekerjaan. Hal ini termasuk:

1. Pengalaman dan keahlian pemrograman komputer.

Sulit untuk membayangkan bagaimana para analisis sistem dapat dengan cukup mempersiapkan bisnis dan spesifikasi teknis untuk programer jika mereka tidak memilki pengalaman programan. Kebanyakan analis system harus menguasai satu atau lebih bahasa pemrograman tingkat tinggi.

1. Pengetahuan umum proses dan teknologi bisnis.

Analis sistem harus mampu berkomunikasi dengan para ahli bisnis untuk memperoleh pemahaman masalah dan kebutuhan mereka. Untuk analis, paling tidak sebagian dari pengetahuan ini datang hanya dari pengalaman. Pada saat yang sama analis yang terinspirasi harus mengambil manfaat dari setiap kesempatan untuk menyelesaikan mata kuliah teori bisnis dasar.

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan didalam tahap ini akan meyebabkan juga kesalahanditahap selanjutnya. Tahap analisa sistem mencakup studi kelayakan analisis kebutuhan.

1. Studi Kelayakan.

Studi kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat dicapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada perusahan serta dampak terhadap lingkungan sekeliling. Tugas-tugas yang tercakup dalam studi kelayakan meliputi:

1. Penentuan masalah dan peluang yang dituju sistem.
2. Pembentukan sasaran sistem baru secara keseluruhan.
3. Pengidentifikasian para pemakai sistem.
4. Pembentukan lingkup sistem.

Selain itu, selama dalam tahapan studi kelayakan sistem analisis juga melakukan tugas-tugas sebagai berikut:

1. Pengusulan perangkat lunak dan perangkat keras untuk sistem baru.
2. Pembuatan analisis untuk membuat atau membeli aplikasi.
3. Pembuatan analisis biaya/manfaat.
4. Pengkajian terhadap resiko proyek.

Studi kelayakan diukur dengan memperhatikan aspek teknologi, ekonomi, faktor organisasi dan kendala hukum, etika, dan yang lain [22].

1. Analisis kebutuhan.

Analisis kebutuhan dilakukan untuk menghasilkan spesifikasi kebutuhan (disebut juga spesifikasi fungsional). Spesifikasi kebutuhan adalah spesifikasi yang rinci tengtang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Spesifikasi ini sekaligus dipakai untuk membuat kesepakatan antara pengembang sistem, pemakai yang kelak akan menggunakan sistem, manajemen, dan mitra kerja yang lain (misalnya auditor inernal).

Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran, volume data yang akan ditangani sistem, jumlah pemakai dan kategori pemakai, serta kontrol terhadap sistem.

Didalam tahap analisis ini sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem, yaitu sebagai berikut:

1. *Identify,* yaitu mengidentifikasi masalah.

Mengidentifikasi (mengenai) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah (*problems*) dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Tahap indentifikasi sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Tahap identifikasi masalah sangat penting karena akan menentukan keberhasilan pada langkah-langkah selanjutnya.

1. *Understand,* yaitu memahami kerja dari sistem yang ada.

Langkah kedua dari tahap analisis sistem adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

1. *Analyze,* yaitu menganalisis sistem tanpa report.

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan.

1. *Report,* yaitu membuat laporan hasil analisis.

Tujuan utama dari pembuatan laporan hasil dilakukan;

1. Pelaporan bahwa analisi telah selesai dilakukan.
2. Meluruskan kesalah pengertian mengenai apa yang telah ditemukan dan dianalisis oleh analis sistem tetapi tidak sesuai menurut manajemen

### **Desain Sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapat gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaiamana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*system design*).

Whitten, et, al. [21] mengungkapkan:” *System design* adalah spesifikasi atau instruksi solusi yang teknis dan berbasis komputer untuk persyaratan bisnis yang diidentifikasikan dalam analisis sistem.”

Desain sistem adalah spesifikasi atau intruksi solusi yang teknis dan berbasis komputer untuk persyaratan bisnis yang diidentifkasikaan dalam analisis sistem.

Driver teknologi sekarang (dan dimasa depan) paling berimpak pada proses dan keputusan desain sistem. Banyak organisasi mengidentifikasikan arsitektur teknologi informasi umum yang didasarkan pada driver-driver teknologi ini.

Tahap desain sistem mempunyai dua tujuan utama, yaitu:

1. Untuk memenuhi kebutuhan kepada pemakai system.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemogram computer dan ahli-ahli teknik lainnya.

Perancangan sistem adalah suatu keinginan membuat desain teknis yang berdasarkan evaluasi yang dilakukan pada kegiatan analisis. Perancangan disini dimaksudkan sebagai proses pemahaman dan perancangan suatu sistem berbasis komputer yang akan menghasilkan komputerisasi.

Dengan demikian, suatu kegiatan perancangan sistem bertujuan untuk menghasilkan suatu sistem komputerisasi. Komputerisasi adalah suatu kegiatan atau sistem pengolahan data dengan menggunakan komputersebagai alat bantu. Perancangan sistem dilakukan setelah tahap analisis sistem selesai dilaksanakan yang kemudian akan menghasilkan output berupa kebutuhan yang akan dijadikan dasar untuk merancang sistem tersebut.

Perancangan sistem terbagi dua, yaitu:

1. Perancangan konseptual.

Perancangan konseptual sering kali disebut dengan perancangan logis. Pada perancangan ini, kebutuhan pemakai dan pemecahan masalah yang teridentifikasi selama tahap analisis sistem mulai dibuat untuk diimplementasikan. Ada tiga langkah penting yang dilakukan dalam perancangan konseptual, yaitu evaluasi alternatif rancangan, penyiapan spesifikasi rancangan, dan penyiapan laporan rancangan sistem secara konseptual.

Menurut Romney, et al. 1997 dalam abdul kadir [23] evaluasi yang dilakukan mengandung hal-hal berikut:

1. Bagaiamana alternatif-alternatif tersebut memenuhi sasaran sistem dan organisasi dengan baik?
2. Bagaimana alternatif-alternatif tersebut memenuhi kebutuhan pemakai dengan baik?
3. Apakah alternatif-alternatif tersebut layak secara ekonomi?
4. Apa saja keuntungan dan masing-masing?

Setelah alternatif rancangan dipilih, tahap selanjutnya adalah penyiapan spesikasi rancangan, yang mencakup elemen- elemen sebagai berikut:

1. Keluaran.

Rancangan laporan mencakup frekuensi laporan (harian, mingguan, dsb), isi laporan, dan laporan cukup ditampilkan pada layar atau perlu dicetak.

1. Penyiapan data.

Dalam hal ini, semua data yang diperlukan untuk membentuk laporan ditentukan lebih detail, termasuk ukuran data dan letaknya dalam berkas.

1. Masukan.

Rancangan masukan meliputi data yang perlu dimasukan kedalam sistem.

1. Prosedur pemrosesan dan operasi.

Rancangan ini menjelaskan bagaimana data dimasukan diproses dan disimpan dalam rangka untuk menghasilkan laporan.

1. Perancangan fisik.

Pada perancangsn ini, rancangan yang masih bersifat konsep diterjemahkan dalam bentuk fisik sehingga terbentuk spesifikasi lengkap tentang modul sistem dan antarmuka antar modul, serta rancangan basis data secara fisik.

Beberapa hasil akhir setelah tahap perancangan fisik berakhir:

1. Rancangan keluaran.

Rancangan keluaran berupa bentuk laporan dan rancangan dokumen

1. Rancangan masukan.

Rancangan masukan berupa rancangan layar untuk pemasukan data.

1. Rancangan antarmuka pemakai dengan sistem.

Rancangan ini berupa rancangan interaksi antara pemakai dan sistem. Misalnya: berupa menu, ikon, dan lain-lain.

1. Rancangan *platform.*

Rancangan ini berupa rancangan yang menentukan *hardware* (perangkat keras) dan *software* (perangkat lunak) yang akan digunakan.

1. Rancangan ini berupa rancangan-rancangan berkas dalam basis data, termasuk penentuan kapasitas masing-masing.
2. Rancangan modul.

Rancangan ini berupa rancangan program yang dilengkapi dengan algortima (cara modul/program bekerja).

1. Rancangan control.

Rancangan ini berupa rancangan kontrol-kontrol yang dugunakan dalam sistem seperti validasi, otorisasi, audit data.

1. Dokumentasi.

Berupa hasil dokumentasi hingga tahap perancangan fisik.

1. Rencana pengujian.

Berupa rencana yang dipakai untuk menguji sistem.

1. Rencana konversi.

Berupa rencana untuk menerapkan sistem baru terhadap sistem lama.

Dalam perancangan sistem yang baik melalui tahapan-tahapan sebagai berikut:

1. Identifikasi masalah yaitu mengidentifikasi masalah yang ada secara rinci agar tidak timbul masalah lain selain masalah utama.
2. Menentukan input, proses dan output yang diinginkan yaitu menginginkan hasil dari perancangan sistem yang dibuat sesuai dengan prosedur.
3. Menentukan algoritma.
4. Mengimplementasikan dengan bahasa pemograman tertentu.
5. Desain sistem dapat dibagi dua bagian, yaitu desain sistem secara umum (general system design) dan desain sistem terinci (*detailed system design*).

### **Desain Sistem Secara Umum**

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada user tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analisis sistem untuk mengidentifikasikan komponen-komponen sistem informasi yang akan didesain secara rinci oleh pemograman komputer dan ahli teknik lainya.

Pada tahap ini komponen-komponen sistem informasi di rancang untuk dikomunikasikan kepada user. Komponen sistem informasi yang didesain adalah model, output - input, database, teknologi dan kontrol.

### **Desain Sistem Terinci (*Detailed* system *design*)**

1. Desain Output Terinci

Desain output terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk output-output dari sistem yang baru. Desain Output Terinci terbagi atas dua, yaitu desain output berbentuk laporan di media kertas dan desain output dalam bentuk dialog di layar terminal.

1. Desain Output dalam bentuk laporan

Desain ini dimaksudkan untuk menghasilkan output dalam bentuk laporan di media kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

1. Desain Output dalam bentuk dialog layar terminal.

Desain ini merupakan rancangan bangun dari percakapan antara pemakai sistem (user) dengan komputer. Percakapan ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan output informasi kepada user, atau keduanya.

Beberapa strategi dalam membuat layar dialog terminal:

1. Dialog pertanyaan/jawaban.
2. Menu.

Menu banyak digunakan karena merupakan jalur pemakai yang mudah dipahami dan mudah digunakan. Menu berisi beberapa alternatif atau option atau option atau pilihan yang di sajikan kepada user. Pilihan menu akan lebih baik bila dikelompokan fungsinya.

1. Desain input Terinci.

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan loleh organisasi. Data hasil dari transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukkan. Desain *Inpu*t terinci dimulai dari desain dokumen dasar tidak didesain desain dengan baik, kemungkinan input yang tercatat dapat salah bahkan kurang

Fungsi dokumen dasar dalam penanganan arus data:

1. Dapat menunjukan macam dari data yang harus dikumpulkan dan ditangkap.
2. Data dapat dicatat dengan jelas, konsisten dan akurat.
3. Dapat mendorong lengkapnya data, disebabkan data yang dibutuhkan disebutkan satu persatu di dalam dokumen dasarnya.
4. Desain Database Terinci.

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpan luar komputer dan digunakan perangkat lunak tertentu untuk manipulasinya. Databse merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagian penyedia informasi bagi para pemakainya. penerapan database dalam sistem informasi disebut database system.

Sistem basis data (database system) adalah suatu sistem informasi yang menginteregasikan kumpulan dari data yang saling berhubungan dengan yang lainya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam dialam suatu organisasi. Dengan sistem basis data ini tiap-tiap orang atau bagian dapat memandang database dari beberapa sudut pandang yang berbeda. Bagian kredit dapat memandangnya sebagai data penjualan, bagian personalia dapat memandangnya sebagai data karyawan, bagian gudang data yang dapat memandangnya sebagai data persediaan. semuanya terintegrasi dalam sebuah data yang umum.

1. Desain Teknologi.

Tahap desain terbagi atas dua yaitu desain teknologi secara umum di rinci. Pada tahap ini kita menentukan teknologi yang akan di pergunakan dalam menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi yang di maksud meliputi:

1. Perangkat keras (*hardware*), yang terdiri dari alat masukan, alat pemroses, alat output dan simpanan luar.
2. Perangkat lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (*operating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak (*application software*)

Sumber daya manusia (*brainware*), misalnya operator komputer, pemrogram, spesialis telekomunikasi, sistem analis dan lain sebagainya. Desain teknologi sangat di perlukan pada tahap implementasi dan pengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

1. Tahap Desain

Tahap desain terbagi menjadi dua, yaitu desain model secara umum dan terinci. Tahap desain model secara umum berupa desain sistem secara fisik dan logika. Desain fisik dapat di gambarkan dengan bagian alir sistem bagian alir dokumen, dan desain secara logika digambarkan dengan diagram dengan arus data(DAD), pada tahap desain model terinci, modelakan didefinisikan secara terinci. urut-urutan langkah proses ini diwakili oleh suatu program komputer.

1. Tahap Desain

Tahap desain terbagi menjadi dua, yaitu desain model secara umum dan terinci. Tahap desain model secara umum berupa desain sistem secara fisik dan logika. Desain fisik dapat di gambarkan dengan bagian alir sistem bagian alir dokumen, dan desain secara logika digambarkan dengan diagram dengan arus data(DAD), pada tahap desain model terinci, modelakan didefinisikan secara terinci. urut-urutan langkah proses ini diwakili oleh suatu program komputer.

**Tabel 2.12**: Bagan Alir Sistem

|  |  |  |  |
| --- | --- | --- | --- |
| No | NAMA SIMBOL | SIMBOL | KETERANGAN |
| 8.  9.  10.  11.  12.  13.  14.  15.  16. | Simbol Pengurutan Offline  Simbol Pita Magnetik  Simbol Hard Disk  Simbol Diskette  SimbolDrum Magnetik  Simbol Pita Kertas Berlubang  Simbol Keyboard  Simbol Display  Simbol Pita Kontrol |  | Menunjukkan proses urut data di luar proses komputer. operasi luar, menunjukkan operasi yang dilakukan di luar proses operasi komputer  Menunjukkan input dan output menggunakan pita *magnetic*.  Menunjukkan *input* dan *output* menggunakan *harddisk*  Menunjukkan *input* dan *output* menggunakan *diskette*  Menunjukkan *input* dan *output* menggunakan drum magnetik  Menunjukkan *input* dan *output* menggunakan pita kertas berlubang.  Menunjukkan *input* yang menggunakan *on-line keyboard.*  Menunjukkan *output* yang ditampilkan di monitor.  Menunjukkan penggunaan pita kontrol (*control tape*) dalam *batch control* total untuk pencocokan di proses *batch processing*. |

|  |  |  |  |
| --- | --- | --- | --- |
| NO | NAMA SIMBOL | SIMBOL | KETERANGAN |
| 17.  18.  19.  20. | Simbol Hubungan Komunikasi  Simbol Garis Alir  Simbol Penjelasan  Simbol Penghubung |  | Menunjukkan proses transmisi data melalui channel komunikasi.  Menunjukkan arus dari proses  Menunjukkan penjelasan dari suatu proses  Menunjukkan penghubung ke halaman yang masih sama atau ke halaman yang lain |

Sumber : Jogyanto, [24].

Untuk mempermudah penggambaran suatu sistem yang ada atau sistem yang baru yang akan dikembangkan secara logika tanpa memperhatikan lingkungan fisik di mana data tersebut mengalir atau lingkungan fisik di mana data tersebut akan disimpan, maka digunakan Diagram Arus Data (DAD) atau *Data Flow Diagram* (DFD). Dalam menggambarkan sistem perlu dilakukan pembentukan simbol, berikut ini simbol-simbol yang sering digunakan dalam DAD :

1. *External entity* (kesatuan luar) atau *boundary* (batas sistem).

Setiap sistem pasti mempunyai batas sistem *(boundary)* yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar *(external entity)* merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan *input* serta menerima *output* dari sistem [24].

**Gambar 2.6:** Notasi kesatuan luar di DAD

1. *Data flow* (arus data).

Arus data ini menunjukkan arus atau aliran data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem [24].

**Gambar 2.7:** Nama Arus Data di DAD

1. *Process* (proses).

Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses [24].



**Gambar 2.8**: Notasi Proses di DAD

1. *Data store* (simpanan data).

Simpanan data pada DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup disalah satu ujungnya [24].

Media Nama Data store

**Gambar 2.9:** Notasi Simpanan Data di DAD

### **Pengujian**

Pada pendekatan berorientasi objek, pengujian merupakan suatu persoalanyang lebih kompleks dibanding dengan pendekatan konvensional, karenakeberadaan pewarisan, polymorphism, dan pengkapsulan pada pengembangansistem berorientasi objek menimbulkan suatu persoalan yang baru untukperancangan kasus pengujian dan analisis hasil.

Hariyanto [25] mengungkapkan bahwa: fitur-fitur berikut berpengaruh dalam teknik-teknik pengujian yang perlu dilakukan:

* Pengkapsulan (*encapsulation*)
* Penyusunan objek-objek (*object composition*)
* Pewarisan (*inheritance*)
* Interaksi (*interaction*)
* *Polymorphism*
* Pengikatan dinamis (*dynamic binding*)
* Guna ulang (*reuse*)
* *Genericity* dan kelas abstrak

Dari kompleksnya fitur –fitur yang mempengahuhi dalam pengujian sistemberorientasi objek maka strategi pengujian dilakukan pada:

1. Pengujian unit, dimana pengujian unit dilakukan hingga beberapa level dengan alasan adanya konsep pewarisan. Pengujian unit ini bertujuan untuk menjamin setiap unit memenuhi spesifikasi. Kelas-kelas merupakan sasaran pengujian unit.
2. Pengujian integrasi, pengujian ini dilakukan untuk memverifikasi implementasi dari satu use case yang telah bekerja seperti yang diharapkan. Pengujian validitas, pengujian ini dilakukan untuk menjamin fungsi-fungsi sistem/aplikasi telah dilakukan secara benar, pengujian di eksekusi ketika satu sistem (subsistem) yang lengkap telah di rakit. Pengujian validasi ini meliputi rincian-rincian objek yang tidak tampak, fokus pada masukan dan keluaran yang tampak oleh pemakai.

### **Implementasi Sistem**

Tahapan implementasi merupakan tahap dimana dilakukan transformasi/ penerjemahan dari bahasa modeling ke suatu bahasa pemrograman. hal ini merupakan tugas dari pemprogram, pada pengembangan sistem/perangkat lunak berorientasi objek penerjemahan dari setiap diagram-diagram DFD yang telah di rancang pada tahap analisis dan desain harus diterjemahkan ke dalam bahasa pemrograman sama persis dengan diagram-diagram yang ada guna menghindari terjadinya perubahan fungsi/tujuan dari pengembangan sistem/perangkat lunak.

### **White Box Testing**

*White Box Testing* atau pengujian *glass box* adalah metode desain *test case* menggunakan struktur kontrol desain prosedural untuk mendapatkan *test case.* Dengan menggunakan metode *White Box* analisis sistem akan memperoleh Test Case yang:

1. Menjamin seluruh *Independent Path* di dalam modul yang dikerjakan sekurang-kurangnya sekali.
2. Mengerjakan seluruh keputusan logical
3. Mengerjakan seluruh *loop* yang sesuai dengan batasannya
4. Mengerjakan seluruh struktur data internal yang menjamin validitas

Untuk melakukan proses pengujian *Test Case* terlebih dahulu dilakukan penerjemahan *flowchart* kedalam notasi *flowgraph* (aliran kontrol). Ada beberapa cara istilah saat pembuatan *flowgraph,* yaitu:

1. *Node* yaitu lingkaran pada *flowgraph* yang menggambarkan satu atau lebih perintah prosedural.
2. *Edge* yaitu tanda panah yang menggambarkan aliran kontrol dari setiap *node* harus mempunyai tujuan *node.*
3. *Region* yaitu daerah yang dibatasi oleh *node* dan *edge* dan untuk menghitung daerah diluar *flowgraph* juga harus dihitung.
4. *Predicate Node* yaitu kondisi yang terdapat pada *node* dan mempunyai karakteristik dua atau lebih *edge* lainnya.
5. *Cyclomatic Complexity* yaitu metrik perangkat lunak yang menyediakan ukuran kuantitaf dari kekompleksan logikal program dan dapat digunakan untuk mencari jumlah path dalam suatu *flowgraph.*
6. *Independen Path* yaitu jalur melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.

Rumus-rumus untuk menghitung jumlah *Independen Path* dalam suatu *flowgraph* yaitu:

1. Jumlah *region* *flowrgaph* mempunyai hubungan dengan *Cyclomatic Complexity (CC).*
2. V(G) untuk *flowgraph* dapat dihitung dengan rumus :
3. V(G) = E – N + 2

Dimana :

E = Jumlah *edge* pada *flowrgaph*

N = Jumlah *node* pada *flowrgaph*

1. V(G) = P + 1

Dimana :

P = Jumlah *predicate node* pada *flowrgaph*

Teknik pelaksanaan pengujian *White Box* ini mempunyai tiga langkah yaitu:

1. Menggambar *flowgraph* yang ditransfer oleh flowchart
2. Menghitung *Cylomatic Complexity* untuk *flowgraph* yang telah dibuat
3. Menentukan jalur pengujian dari *flowgraph* yang berjumlah sesuai dengan *Cyclomatic Complexity* yang telah ditentukan.

2

4

5

8

7

1

3

6

10

11

9

**Gambar 2.10:** Bagan Air

(Sumber: Roger S. Pressman, [26])

Bagan alir digunakan untuk menggambarkan struktur kontrol program dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut simpul grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut edges atau links, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. Edge harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural.

Simpul Predikat

Edge

Node

Region

R1

R4

R2

R3

**Gambar 2.1:** Flowgraph

(Sumber: Roger S. Pressman, [26]).

Dari gambar *flowgraph* di atas didapat:

Path 1 = 1 – 11

Path 2 = 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Path 3 = 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path 4 = 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis set untuk diagram alir.

*Cyclomatic complexity* digunakan untuk mencari jumlah path dalam satu flowgraph. Dapat dipergunakan rumusan sebagai berikut:

1. Jumlah region grafik alir sesuai dengan *cyclomatic complexity*.
2. *Cyclomatix complexity* V(G) untuk grafik alir dihitung dengan rumus:

V(G) = E – N + 2 …………. (1)

Dimana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

3. *Cyclomatix complexity* V(G) juga dapat dihitung dengan rumus:

V(G) = P + 1 …… (2)

Dimana P = jumlah *predicate node* pada grafik alir

Dari Gambar di atas dapat dihitung *cyclomatic complexity*:

1. *Flowgraph* mempunyai 4 region

2. V(G) = 11 *edge* – 9*node* + 2 = 4

3. V(G) = 3 *predicate node* + 1 = 4

Jadi *cyclomatic complexity* untuk *flowgraph* adalah 4

*Cyclomatic Complexity* yang tinggi menunjukkan prosedur kompleks yang sulit untuk dipahami, diuji dan dipelihara. Ada hubungan antara *Cyclomatic Complexity* dan resiko dalam suatu prosedur.

### **Black Box Testing**

Menurut Pressman [26] *Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemutusan kesalahan

Tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan berikut ini:

1. Bagaimana validitas fungsional diuji?
2. Bagaimana perilaku dan kinerja sistem diuji?
3. Apa kelas *input* akan membuat kasus uji yang baik?
4. Apakah sistem *sensitive* terhadap nilai input tertentu?
5. Bagaimana batas-batas kelas data yang terisolasi?
6. Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
7. Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi?
8. Ciri-Ciri Black Box Testing
9. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
10. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.
11. *Black box testing* melakukan pengujian tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing, specification-based testing, input/output testing* atau *functional testing*
12. Jenis teknik *design* tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan.
13. *Equivalence Class Partitioning*
14. *Boundary Value Analysis*
15. *State Transitions Testing*
16. *Cause-Effect Graphing*
17. Kategori *error* yang akan diketeahui melalui *black box testing*
18. Fungsi yang hilang atau tak benar
19. *Error* dari antar-muka
20. *Error* dari struktur data atau akses eksternal database
21. *Error* dari kinerja atau tingkah laku
22. *Error* dari inisialisasi dan terminasi

### **Perangkat Lunak Pendukung**

Perangkat lunak pendukung yang digunakan penulis dalam membangun sistem ini yaitu PHP dan MySQL, seperti pada tabel di bawah ini:

**Tabel 2.14:** Perangkat Lunak Pendukung

|  |  |  |
| --- | --- | --- |
| **NO** | **TOOLS** | **KEGUNAAN** |
| 1 | PHP | Sebuah bahasa *scripting* yang terpasang pada HTML. Yang bertujun untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. |
| 2 | MySQL | Salah satu pengolah database yang menggunakan SQL (*Strukture Query Language*) sebagai bahan dasar untuk mengakses databasenya. Yang memiliki keuntungan seperti *open source* dan memiliki kemampuan menampung kapasitas yang besar. |

## 

## **Kerangka Pikir**

Masalah

1. Bagaimana hasil penerapan metode K-NN untuk klasifikasi tingkat penyebaran penyakit masyarakat Kota Gorontalo?
2. Seberapa besar tingkat akurasi metode K-NN untuk klasifikasi tingkat penyebaran penyakit masyarakat di Kota Gorontalo?

Identifikasi Pola

Observasi

Pengumpulan Dataset

Clasifikasi

*Confusion Matriks*

Sistem Development

Analisis Sistem

Diagram Konteks (DFD)

Diagram Berjenjang (DFD)

Diagram Arus Data Level 0, dst (DFD)

Kamus Data

Desain output

Desain Input

Desain Basis Data (Struktur Data)

Desain Teknologi

Desain Sistem

Konstruksi Sistem

Programming (PHP), Database (MYSQL)

DataBase (MYSQL)

Pengujian Sistem

Program (White Box)

Interface (Black Box)

1. Mengklasifikasi tingkat penyebaran penyakit masyarakat di Kota Gorontalo menggunakan metode K-NN.
2. Untuk memperoleh akurasi yang baik dalam melakukan penerapan metode K-NN pada klasifikasi tingkat penyebaran penyakit masyarakat di Kota Gorontalo.

Tujuan

**Gambar 2.12:** Bagan Kerangka Pikir