**BAB IV**

**ANALISA DAN PERANCANGAN**

* 1. **Analisa Sistem**

Analisa sistem adalah penguraian dari suatu sistem informasi kedalam bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Analisa sistem merupakan tahap awal dalam perancangan dan pengembangan sebuah sistem yang akan dirancang, karena tahap inilah akan diukur dan dievakuasi tentang kinerja dari sistem yang dirancang. Identifikasi terhadap masalah-masalah yang ada dan langkah-langkah untuk kebutuhan perancangan yang diharapkan.

Dalam melakukan analisis sistem terlebih dahulu harus mengetahui dan memahami sistem, untuk menganalisa system diperlukan data dari sistem untuk dianalisa. Data yang diperlukan adalah hal-hal yang diperlukan untuk definisi data.

**4.1.1 Analisa Data**

Analisa data merupakan tahap untuk melakukan penganalisaan terhadap data-data yang dibutuhkan untuk perancangan sistem yang akan dibuat, dalam hal ini penulis mengambil data melalui *literature-literature* yang berhubungan dengan tema penelitian, untuk mencari informasi menyusun teori-teori yang berhubungan dengan pembahasan sehingga terjadi perpaduan yang komplek antara yang satu dengan yang lainnya.

**Tabel 4.1 data Penjualan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Kode Barang | Jenis Barang | Merek | Tahun | Harga | Status |
| 1 | 3322 | Laptop | Axioo | 2004 | 7500000 | Laris |
| 2 | 5651 | PC | Toshiba | 2009 | 12000000 | Laris |
| 3 | 8819 | PC | Toshiba | 2010 | 13000000 | Laris |
| 4 | 1328 | PC | Toshiba | 2009 | 12000000 | Laris |
| 5 | 1376 | PC | Axioo | 2010 | 13500000 | Laris |
| 6 | 5513 | Laptop | Acer | 2012 | 9000000 | Laris |
| 7 | 7074 | Laptop | Toshiba | 2003 | 6500000 | Tidak Laris |
| 8 | 6005 | Laptop | Toshiba | 2005 | 12000000 | Tidak Laris |
| 9 | 1329 | PC | Axioo | 2010 | 13000000 | Laris |
| 10 | 6646 | Laptop | Axioo | 2008 | 11000000 | Laris |
| 11 | 6469 | Laptop | Acer | 2011 | 19000000 | Tidak Laris |
| 12 | 6336 | PC | Toshiba | 2010 | 13000000 | Laris |
| 13 | 1241 | Laptop | Acer | 2007 | 15000000 | Tidak Laris |
| 14 | 2770 | PC | Toshiba | 2010 | 13000000 | Laris |
| 15 | 6322 | Laptop | Toshiba | 2003 | 7000000 | Tidak Laris |
| 16 | 2431 | PC | Toshiba | 2009 | 12000000 | Laris |
| 17 | 2212 | Laptop | Axioo | 2008 | 11000000 | Laris |
| 18 | 1174 | Laptop | Acer | 2004 | 10000000 | Laris |
| 19 | 5535 | PC | Axioo | 2009 | 12000000 | Laris |
| 20 | 1337 | PC | Toshiba | 2010 | 13000000 | Laris |
| 21 | 7334 | Laptop | Acer | 2004 | 10000000 | Laris |
| 22 | 1244 | Laptop | Axioo | 2005 | 9000000 | Laris |
| 23 | 8005 | PC | Toshiba | 2010 | 13000000 | Laris |
| 24 | 6221 | PC | Axioo | 2009 | 12000000 | Laris |
| 25 | 1021 | Laptop | Acer | 2007 | 15500000 | Tidak Laris |

Proses klasifikasi tiap-tiap *field* tabel Penjualan

1. Klasifikasi Merek

**Tabel 4.2 Klasifikasi Merek**

|  |  |  |
| --- | --- | --- |
| Klasifikasi Pabrikan | Laris | Tidak Laris |
| Axioo | 8 | 0 |
| Toshiba | 8 | 3 |
| Acer | 3 | 3 |

1. Klasifikasi Jenis Barang

**Tabel 4.3 Klasifikasi Jenis Barang**

|  |  |  |
| --- | --- | --- |
| Klasifikasi Jenis | Laris | Tidak Laris |
| PC | 12 | 0 |
| Laptop | 7 | 6 |

1. Klasifikasi Tahun

**Tabel 4.4 Klasifikasi Tahun**

|  |  |  |  |
| --- | --- | --- | --- |
| Klasifikasi Tahun | Kode | Laris | Tidak laris |
| 2002-2006 | A | 5 | 3 |
| 2007-2011 | B | 14 | 3 |

1. Klasifikasi Harga

**Tabel 4.5 Klasifikasi Harga**

|  |  |  |  |
| --- | --- | --- | --- |
| Klasifikasi Harga | Kode | Laris | Tidak Laris |
| 5000000-10000000 | A | 5 | 2 |
| 10000000-15000000 | B | 14 | 2 |
| 15000000-20000000 | C | 0 | 2 |

**4.1.2 Analisa Proses**

Dalam hal ini penulis mengambil data pada CV Putra Elektronik di kota Dumai, dimana penulis mengambil sampel datanya selama sebulan yaitu pada bulan Februari 2014, data tersebut diketik ulang menggunakan *Microsoft Excel* karena pihak perusahaan memberikan data dalam bentuk *hardcopy*. *Decision tree* memiliki dua jenis *atribut* dari data yang terdiri dari beberapa *atribut* input dari *atribut* target dan tentunya mendukung masalah yag ada, fungsinya sebagai pembanding dalam perhitungan *Gain* dan *Ratio*.

Dalam data training *set atibut sample* dan minimal harus memiliki satu atribut target yang nilainya merupakan kesimpulan sementara permasalahan dari setiap *instance( record*), dalam penelitian ini nilai dari *atribut* target adalah: laris atau tidak laris. *Atribut input* yang memiliki *gain ratio* yang terbesar adalah *atribut* yang menjadi akar. Contohnya seperti gambar di bawah ini :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Kode Barang | Jenis Barang | Merek | Tahun | Harga | Status |
| 1 | 3322 | Laptop | Axioo | 2004 | 7500000 | Tidak Laris |
| 2 | 8819 | PC | Toshiba | 2010 | 13000000 | Laris |

**Atribut Target**

**Atribut Input**

**sampel**

**Gambar 4.1 konsep data dalam *decision tree***

**4.1.3 Perancangan Algoritma**.

Perancangan data mining ini penulis menggunakan algoritma C4.5. Proses pada pohon keputusan adalah mengubah bentuk data (table) menjadi model pohon, mengubah model pohon menjadi rule, dan menyederhanakan rule.

Dalam kasus yang tertera pada gambar 4.1 akan dibuat pohon keputusan untuk menentukan barang yang terjual atau tidak dengan melihat merek, jenis barang, tahun, dan harga. Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut :

1. Pilih atribut akar.
2. Buat cabang untuk tiap-tiap nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses unutk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain,* seharusnya kita harus mencari nilai *entropy* terlebih dahulu.

Rumusnya seperti persamaan berikut.

*Entropy(S)*

Keterangan :

S : himpunan kasus

A : atribut

n : jumlah partisi S

pi : proporsi dari Si terhadap S

Sementara itu, penghitungan nilai *gain* dapat dilihat pada persamaan 2 berikut.

*Gain(S, A) = Entropy(S)-*

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|Si| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

Berikut ini adalah penjelasan lebih terperinci mengenai tiap-tiap langkah dalam pembentukan pohon keputusan dengan menggunakan algoritma C4.5 untuk menyelesaikan permasalahan pada tabel 4.1.

1. Menghitung jumlah kasus untuk keputusan laris, jumlah kasus untuk keputusan tidak laris, Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut jumlah barang, barang terjual, dan sisa barang. Setelah itu, lakukan penghitungan gain untuk setiap atribut. Hasil perhitungan ditunjukkan oleh tabel 4.6.

**Tabel 4.6 Perhitungan Node 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NODE |  |  | Jumlah Kasus (S) | Tidak laris (S1) | Laris  (S2) | Entropy | Gain |
| 1 | TOTAL |  | 25 | 6 | 19 | 0.795040 |  |
|  | Harga |  |  |  |  |  |  |
|  |  | A | 7 | 2 | 5 | 0,868721 |  |
|  |  | B | 16 | 2 | 14 | 0.543564 |  |
|  |  | C | 2 | 2 | 0 | 0 |  |
|  | Merek |  |  |  |  |  | 0.180489 |
|  |  | Axioo | 8 | 0 | 8 | 0 |  |
|  |  | Toshiba | 11 | 3 | 8 | 0.851252 |  |
|  |  | Acer | 6 | 3 | 3 | 1.000000 |  |
|  | Jenis |  |  |  |  |  |  |
|  |  | PC | 12 | 0 | 12 | 0 |  |
|  |  | Laptop | 13 | 6 | 7 | 0.995378 |  |
|  | Tahun |  |  |  |  |  | 0.143717 |
|  |  | A | 8 | 3 | 5 | 0.950672 |  |
|  |  | B | 17 | 3 | 14 | 0.671247 |  |

Baris total *entropy* pada tabel 4.6 dihitung dengan persamaan 1 sebagai berikut :

*Entropy(Total) =*)+)= 0.795040

*Entropy(Total) =*0.795040"

***Entropy*** (Total ***,*** Harga A) =)+)= 0,868721

***Entropy*** (Total , Harga B) = )+)= 0.543564

***Entropy*** (Total , Harga C) =)+) =0

***Entropy* Pabrikan**

***Entropy*** (Total ***,***Axioo) =)+) = 0

***Entropy*** (Total ***,***Toshiba) =)+) = 0.851252

***Entropy*** (Total ***,*** Acer) =)+) = 1

***Entropy* Jenis**

***Entropy*** (Total ***,*** PC) =)+) = 0

***Entropy*** (Total ***,*** Laptop) =)+) = 0.995378

***Entropy* Tahun**

***Entropy*** (Total, Tahun A) =)+) =0.950672

***Entropy*** (Total ***,***Tahun B) =)+) = 0.680077

***Gain*** (Total, Harga) = *Gain(S, A) = Entropy(S)-*

***Gain***(Total,Harga) = 0.795040279*-*

***Gain***(Total,Merek) = 0.795040279*–*

= 0.180489

***Gain***(Total,Jenis) *=* 0.795040 *–*

***Gain***(Total,Tahun) = 0.795040 *–*

= 0.143717

Dari hasil tabel 4.6 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah jenis, yaitu . dan nilai atribut PC sudah mengklasifikasikan kasus menjadi laris, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut Laptop masih perlu dilakukan perhitungan lagi. dengan demikian Laptop dapat menjadi node cabang dari nilai atribut Jenis.

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada gambar 4.2.

Laris

Laptop PC

**Gambar 4.2 Pohon Keputusan Hasil Perhitungan Node 1**

1. Menghitung jumlah kasus, jumlah kasus untuk keputusan laris, jumlah kasus untuk keputusan tidak laris, dan *entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut harga, merek, dan tahun yang dapat menjadi node akar dari nilai atribut adalah Laptop. Setelah itu, lakukan penghitungan *Gain* untuk tiap-tiap atribut. Hasil perhitungan ditunjukkan oleh tabel 4.7.

**Tabel 4.7 Perhitungan Node 1.1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NODE |  |  | Jumlah Kasus (S) | Tidak laris (S1) | Laris  (S2) | Entropy | Gain |
| 1.1 | Jenis-Laptop |  | 13 | 6 | 7 | 0.995727 |  |
|  | Harga |  |  |  |  |  | 0.459566 |
|  |  | A | 7 | 4 | 3 | 0.985815 |  |
|  |  | B | 4 | 0 | 4 | 0.000000 |  |
|  |  | C | 2 | 2 | 0 | 0.000000 |  |
|  | Merek |  |  |  |  |  | 0.534188 |
|  |  | Acer | 6 | 3 | 3 | 1 |  |
|  |  | Axioo | 4 | 0 | 4 | 0 |  |
|  |  | Toshiba | 3 | 3 | 0 | 0 |  |
|  | Tahun |  |  |  |  |  | 0.032720 |
|  |  | A | 8 | 3 | 5 | 0.958042 |  |
|  |  | B | 5 | 3 | 2 | 0.970950 |  |

Baris total *entropy* pada tabel 4.7 dihitung dengan persamaan sebagai berikut.

***Entropy*** (Total ***,*** Laptop) =)+) = 0.995727

***Entropy* Harga**

***Entropy*** (Laptop ***,*** Harga A) =) +)= 0.985815

***Entropy*** (Laptop, Harga B) = ) +)= 0

***Entropy*** (Laptop ,Harga C) =) +) = 0

***Entropy* Merek**

***Entropy*** (Laptop ***,***Acer) =) +) = 1

***Entropy*** (Laptop ***,***Axioo) =) +) = 0

***Entropy*** (Laptop ***,*** Toshiba) =) +)= 0

***Entropy* Tahun**

***Entropy*** (Laptop, Tahun A) =)+) = 0.958042

***Entropy*** (Laptop***,*** Tahun B) =)+) = 0.970950

***Gain*** (Laptop, Harga) = *Gain(S, A) = Entropy(S)-n)*

*=* 0.995727**-**

0.459566

***Gain*** (Laptop, Merek) = *Gain(S, A) = Entropy(S)-*

*=* 0.995727**-**

0.534188

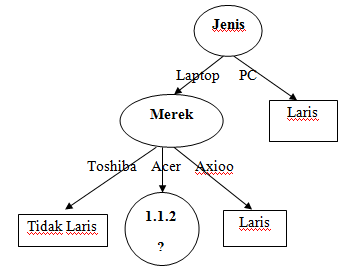
***Gain*** (Laptop, Tahun) = *Gain(S, A) = Entropy(S)-n)*

*=*0.995727**-**

0.032720

Dari hasil pada tabel 4.7 dapat diketahui bahwa atribut dengan Gain tertinggi adalah Merek, yaitu sebesar 0.534188. Ada tiga nilai atribut dari Merek yaitu Axioo, Toshiba dan Acer. Dari ketiga nilai atribut tersebut, nilai atribut Axioo sudah mengklasifikasikan kasus yaitu keputusannya laris dan nilai atribut Toshiba sudah mengklasifikasikan kasus yaitu tidak laris, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut Acer masih perlu dilakukan perhitungan lagi. dengan demikian merek dapat menjadi node cabang dari nilai atribut Laptop.

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada gambar 4.3 berikut.



**Gambar 4.3 Pohon Keputusan Hasil Perhitungan Node 1.1**

1. Menghitung jumlah kasus, jumlah kasus untuk keputusan laris, jumlah kasus untuk keputusan tidak laris, dan *entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut harga, dan tahun yang dapat menjadi node akar dari nilai atribut adalah Merek. Setelah itu, lakukan penghitungan *Gain* untuk tiap-tiap atribut. Hasil perhitungan ditunjukkan oleh tabel 4.8.

**Tabel 4.8 Perhitungan Node 1.1.2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NODE |  |  | Jml Kasus (S) | Tidak laris (S1) | Laris (S2) | Entropy | Gain |
| 1.1.2 | Jenis Laptop dan Merek Acer |  | 6 | 3 | 3 | 1.000000 |  |
|  | Harga |  |  |  |  |  | 1 |
|  |  | A | 3 | 0 | 3 | 0.000000 |  |
|  |  | B | 1 | 1 | 0 | 0.000000 |  |
|  |  | C | 2 | 2 | 0 | 0.000000 |  |
|  | Tahun |  |  |  |  |  | 1 |
|  |  | A | 3 | 0 | 3 | 0.000000 |  |
|  |  | B | 3 | 3 | 0 | 0.000000 |  |

Baris total *entropy* pada tabel 4.8 dihitung dengan persamaan sebagai berikut :

***Entropy* (**Total Acer**)** *=*) +) = 1

***Entropy* Harga**

***Entropy*** (Acer ***,***harga A) =) +) = 0

***Entropy*** (Acer, harga B) = ) +) = 0

***Entropy*** (Acer, harga C) =) +) = 1

***Entropy* Tahun**

***Entropy*** (Acer, Tahun A) =) +) = 0

***Entropy*** (Acer, Tahun B) =) +) = 0

***Gain*** (Acer, Harga) = *Gain(S, A) = Entropy(S)-*

*=*1**-**

1

***Gain*** (Acer,Tahun) = *Gain(S, A) = Entropy(S)-*

*=* 1**-**

1

Dari hasil pada tabel 4.8 dapat diketahui bahwa atribut dengan Gain tertinggi sama, yaitu sebesar 1. dengan demikian harga dapat menjadi node cabang dari nilai atribut Acer. Karena attribut yang lebih berpengaruh. Ada tiga nilai atribut dari harga yaitu A, B dan C. Dari ketiga nilai atribut tersebut, nilai atribut A sudah mengklasifikasikan kasus menjadi laris, nilai atribut B sudah mengklasifikasikan kasus menjadi tidak laris dan nilai attribut C sudah mengklasifikasikan kasus menjadi tidak laris sehingga tidak perlu dilakukan perhitungan lebih lanjut,

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada gambar 4.4 berikut.

Laptop PC

Laris

TidakLaris

Laris

Toshiba Acer Axioo

Laris

Tidak Laris

Tidak Laris

A B C

**Gambar 4.4 Pohon Keputusan Hasil Perhitungan Node 1.1.2**

* 1. **Perancangan**
     1. **Pembuatan Diagram UML**

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Adapun UML yang dirancang dalam pembuatan sistem ini sebagai berikut :

1. **Pembuatan Use Case Diagram**

*Use case diagram* adalah deskipsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Pada *Use Case Diagram* aktor pengguna dihadapkan dengan beberapa fitur menu yang dapat digunakan, dan disini terdapat 1 aktor yaitu, waka kurikulum.

1. Definisi Aktor

Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam system. Pada system yang dibangun untuk memprediksi penjualan pada CV Putra Elektronik ini terdapat satu aktor yaitu Manager. Fungsi aktor tersebut dijelaskan pada tabel berikut :

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Manager | Manager dapat melakukan apa saja dalam sistem seperti tambah setting, lihat mining, tamabah atribut, tampil proses, lihat aturan , tampil kasus, tampil testing |

1. Definisi Use Case

Kegiatan-kegiatan yang akan terjadi di dalam sistem antara para aktor dengan *use case* dapat digambarkan pada table berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Use Case** | **Deskripsi** | **Aktor** |
| 1 | Lihat Home | Manager bisa melihat home ketika masuk ke dalam sistem. | Manager |
| 2 | Lihat Profile | Manager bisa melihat profile ketika masuk ke dalam sistem. | Manager |
|  |  |
| 3 | Lihat Semua Data | Manager dapat melihat semua data. | Manager |
| 4 | Print Semua Data | Manager dapat mencetak semua data. | Manager |
| 5 | Hapus Semua Data | Manager dapat menghapus semua data. | Manager |
| 6 | Input Data Baru | Manager dapat input data penjualan baru pada sistem.. | Manager |
|  |  |
| 7 | Lakukan Mining | Digunakan untuk menampilkan atau proses data-data menggunakan C.45 | Manager |
|  |  |
| 8 | Perhitungan C.45 | Merupakan Hasil Perhitungan C.45 | manager |
| 9 | Hapus hitung C.45 | Digunakan untuk menghapus hasil perhitungan C.45 | Manager |
| 10 | Lihat Pohon Keputusan | Manager dapat melihat pohon keputusan yang di hasilkan dari Perhitungan C.45 | Manager |
| 11 | Hapus Pohon Keputusan | Manager dapat menghapus Pohon keputusan yang di hasilkan. | Manager |
| 12 | Print Pohon Keputusan | Manager dapat mencetak Pohon keputusan yang di hasilkan. | Manager |
| 13 | Login | Manager untuk masuk ke dalam sistem maka terlebh dahulu harus login. | Manager |

Berikut ini adalah gambar *use case diagram* yang terbentuk:

****

**Gambar 4.5 Use Case Diagram**

Pada diagram diatas dapat dilihat ada tiga aktor yang yang dapat berintaksi dengan sistem, dimana aktor yang dimaksud adalah manajer. Manajer disini berperan dalam keseluruhan dari sistem,

1. **Pembuatan Class Diagram**

*Class Diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* pada sistem dapat dilihat pada gambar 4.2.



**Gambar 4.6 Class Diagram**

*Class diagram* diatas menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti hubungan antara data penjualan, manajer dengan sistem.

1. **Struktur Database**

Struktur database merupakan kumpulan dari data-data beserta tipenya yang merupakan komponen penting dalam membuat suatu program. Struktur database merupakan struktur data yang saling berhubungan satu sama lain sehingga sangat diperlukan dalam menjalankan program dan juga menyimpan data dalam suatu sistem database seperti di bawah ini :

1. Nama tabel : Data\_Survey

Primary key : kode\_barang

Fungsi : menyimpan data Penjulan Barang

**Tabel 4.9 Data Survey**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Key | Type | Length | Keterangan |
| Kode Barang |  | VARCHAR | 255 | Kode Barang |
| Jenis\_barang |  | VARCHAR | 255 | Jenis barang |
| Merek |  | VARCHAR | 50 | Merek Barang |
| Tahun |  | VARCHAR | 255 | Tahun dari Barang |
| Harga |  | INTEGER | 10 | Harga barang |
| Status |  | VARCHAR | 255 | Status |

1. Nama tabel : Atribut

Primary key : atribut

Fungsi : untuk menampung data-data atribut

**Tabel 4.10 Atribut**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Key | Type | Length | Keterangan |
| atribut |  | VARCHAR | 50 | Nama\_Atribut |
| Nilai\_atribut |  | VARCHAR | 255 | Isi atau nilai dari atribut |

1. Nama tabel :manager

Fungsi : menyimpan data manager yang login

**Tabel 4.11 manager**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Key | Type | Length | keterangan |
| username |  | VARCHAR | 255 | Username Manager |
| password |  | VARCHAR | 255 | Password manager |
| Nama\_lengkap |  | VARCHAR | 255 | Mana lengkap manajer |
| email |  | VARCHAR | 255 | Email manajer |
| No\_telp |  | VARCHAR | 255 | No telp manager |
| level |  | VARCHAR | 50 | Level manager |

1. Nama tabel : Mining

Fungsi : menyimpan data hasil proses mining

**Tabel 4.12 Mining**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Key | Type | Length | Keterangan |
| atribut |  | VARCHAR | 30 | Nama\_atribut |
| Nilai atribut |  | VARCHAR | 50 | Nilai Atribut |
| Jml\_kasus\_total |  | INT | 5 | Jumlah kasus total |
| Jml\_laris |  | INT | 5 | Jumlah laris |
| Jml\_tdk\_laris |  | INT | 5 | Jumlah tidak laris |
| entropy |  | VARCHAR | 20 | entropy |
| Inf\_gain |  | VARCHAR | 5 | Information gain |
| Inf\_gain\_temp |  | VARCHAR | 5 | information gain temp |
| Split\_info |  | VARCHAR | 5 | Split info |
| Split\_info\_temp |  | VARCHAR | 5 | Split info temporary |
| Gain\_ratio |  | VARCHAR | 5 | Gain ratio |

1. Nama tabel : Pohon Keputusan C45

Fungsi : menyimpan data pohon keputusan dari hasil mining.

**Tabel 4.13 Pohon Keputusan C45**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | key | Type | Length | Keterangan |
| atribut |  | VARCHAR | 20 | Nama dari atribut |
| nilai\_atribut |  | VARCHAR | 30 | Nilai dari atribut |
| Id\_parent |  | VARCHAR | 255 | Id parent |
| Jml Laris |  | NUMERIC | 15 | Jumlah barang laris |
| Jml tdk laris |  | VARCHAR | 30 | Jumlah tidak laris |
| keputusan |  | VARCHAR | 30 | Hasil keputusan |
| diproses |  | INTEGER | 10 | Jumlah di proses |
| Kondisi atribut |  | VARCHAR | 40 | Kondisi atribut |
| Looping kondisi |  | VARCHAR | 50 | Perulangan kondisi |

1. Nama tabel : Iterasi c45

Primary key : id

Fungsi : menyimpan data iterasi c.45

**Tabel 4.14 iterasi c45**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | key | Type | length | Keterangan |
| iterasi |  | INTEGER |  | Iterasi |
| Atribut\_gain\_ratio\_max |  | VARCHAR | 30 | Atribut gain ratio max |
| atribut |  | VARCHAR | 30 | atribut |
| Nilai\_atribut |  | VARCHAR | 30 | Nilai atribut |
| Jml\_kasus\_total |  | INTEGER | 5 | Jumlah kasus total |
| Jml laris |  | INTEGER | 5 | Jumlah laris |
| Jml tdk laris |  | INTEGER | 5 | Jumlah tidak laris |
| entropy |  | VARCHAR | 5 | entropy |
| Info gain |  | VARCHAR | 5 | Info gain |
| Split\_info |  | VARCHAR | 5 | Split info |
| Gain ratio |  | VARCHAR | 5 | Gain ratio |

1. **Pembuatan Sequence Diagram**

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation* bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

1. **Sequence Login Manager**



**Gambar 4.7 Sequence Login manager**

Manajer masuk ke form login dengan memasukkan username dan password terlebih dahulu. Lalu manajer dapat melihat kasus, proses data penjulan yang tersedia pada form tersebut.

1. **Sequence Diagram Kelola Semua Data**



**Gambar 4.8 Sequence Kelola Semua Data**

Manajer masuk ke form semua data dengan melakukan login terlebih dahulu. Lalu manajer dapat melihat dan mengelola semua data yang tersedia pada form tersebut.

1. **Sequence Diagram Perhitungan c.45**



**Gambar 4.9 Sequence perhitungan c.45**

Manajer masuk ke form perhitungan c.45 dengan melakukan login terlebih dahulu. Lalu manajer dapat melihat atribut dan mengetahui hasil dari proses data mining c.45 pada form tersebut.

1. **Collaboration Diagram**

Merupakan cara alternatif untuk menampilkan suatu scenario. Menampilkan interaksi obyek yang terorganisasi di sekitar obyek dan hubungannya dengan obyek yang lain.

1. **Collaboration Login Manager**



**Gambar 4.11 Collaboration Login Manager**

Manajer masuk ke form login dengan memasukkan username dan password terlebih dahulu. Lalu manajer dapat melihat kasus, proses data penjulan yang tersedia pada form tersebut.

1. **Collaboration Kelola Semua Data**



**Gambar 4.12 Collaboration Kelola Semua data**

Manajer masuk ke form daftar aturan dengan melakukan login terlebih dahulu. Lalu manajer dapat melihat daftar aturan dan menyimpan ke excel dari data barang yang tersedia pada form tersebut.

1. **Collaboration Diagram Perhitungan c.45**



**Gambar 4.13 Collaboration Perhitungan c.45**

Manajer masuk ke form testing dengan melakukan login terlebih dahulu. Lalu manajer dapat melihat atribut dan mengetahui hasil dari proses data barang yang tersedia pada form tersebut.

1. **Pembuatan State chart Diagram**
2. **State Chart Diagram Kelola Semua data**

****

**Gambar 4.15 State Chart Kelola semua data**

Diagram diatas menjelaskan bahwa perubahan dari *state* disaatmanajer login, kemudian manajer dapat masuk ke form semua data untuk mengelolanya, dan dapat melakukan setting atribut,melihat dan memproses data barang.

1. **State Chart Diagram Atribut**

****

**Gambar 4.16 State chart Diagram Atribut**

Diagram diatas menjelaskan bahwa perubahan dari *state* disaatmanajer login, kemudian manajer dapat masuk ke manajemen atribut, dan bisa melihat proses yang terjadi dan menyimpan data atribut.

1. **Pembuatan Activity Diagram**

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Adapun *Activity diagram* pada sistem secara umum yaitu dapat dilihat pada gambar 4.19



**Gambar 4.19 Activity Diagram**

Pada diagram *activity* diatas dapat dilihat alir aktivitas semua aktor terhadap sistem. Aktor harus login terlebih dahulu sebelum melakukan aksi.

1. **Deployment Diagram**

Menggambarkan tentang detail hubungan infrastruktur yang terhubung dalam sistem, yaitu di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

Adapun *Deployment Diagram* pada sistem secara umum yaitu dapat dilihat pada gambar 4.20.

****

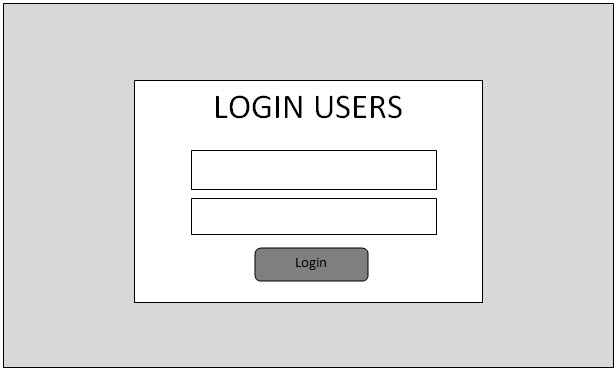
**Gambar 4.20 Deployment Diagram**

* + 1. **Desain User Interface**

Desain user interface merupakan desain tampilan sistem yang akan dibuat, mulai dari desain halaman utama atau index sampai desain menu proses. Adapun desain yang dimaksud adalah sebagai berikut :

1. **Antarmuka Halaman Login**

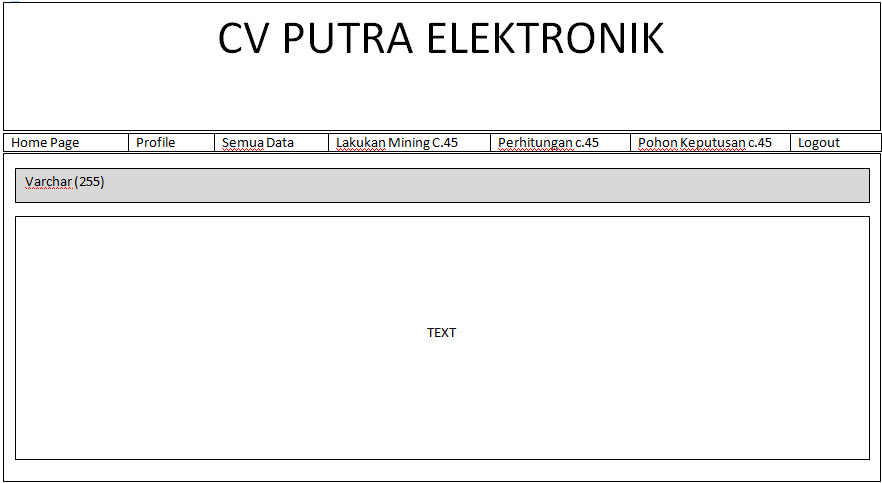
Merupakan halaman yang pertama kali ditampilkan ketika mengakses sistem. Pada halaman utama terdapat terdapat beberapa tombol, diantaranya adanya tombol login untuk akses masuk. Halaman utama dapat digambarkan seperti Gambar 4.21.



**Gambar 4.21 Perancangan Antarmuka Login**

1. **Antarmuka Halaman Utama**

Halaman ini merupakan halaman pertama muncul saat pengguna membuka aplikasi. Perancangan halaman awal dapat dilihat pada gambar 4.22 dibawah ini.



**Gambar 4.22 Perancangan Antarmuka Halaman Utama**

1. **Antarmuka Halaman Semua Data Penjualan**

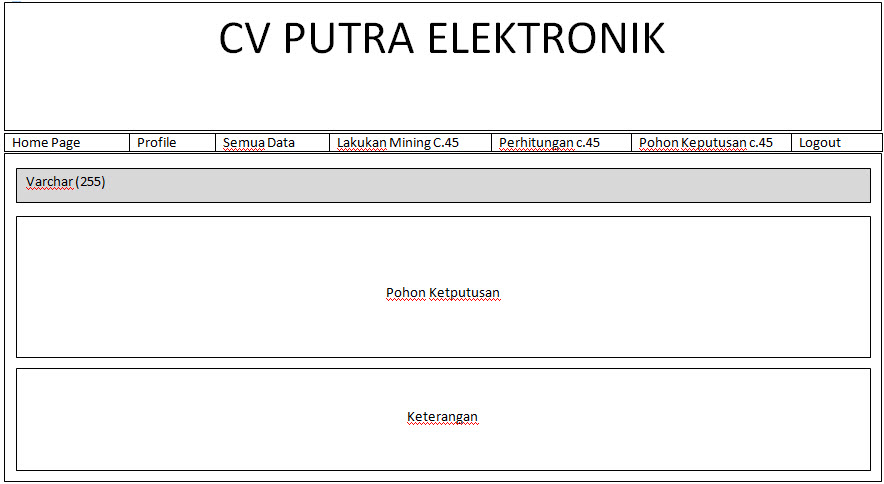
Halaman ini merupakan halaman yang akan muncul saat pengguna memilih menu semua data. Halaman ini digunakan untuk menginputkan atribut, melihat kasus dari atribut yang telah di tentukan, yaitu sebagai berikut :



**Gambar 4.23 Perancangan Semua Data Penjualan**

1. **Antarmuka Halaman Form Pohon Keputusan**

Halaman ini merupakan halaman yang akan muncul saat pengguna memilih menu Pohon Keputusan. Halaman ini digunakan untuk meihat hasil pohon keputusan yang di ciptakan dari proses mining.



**Gambar 4.24 Perancangan Form Pohon Keputusan**

1. **Antarmuka Halaman Form Input Data Penjualan**

Halaman ini merupakan halaman yang akan muncul saat pengguna memilih menu tambah data penjualan. Halaman ini digunakan untuk menambahkan data penjualan baru pada proses mining.



**Gambar 4.25 Perancangan Form Input Data Penjualan**