**BAB III**

**Analisa dan Desain**

1. **Analisis Permasalahan**

Kami telah melakukan serangkaian analisis terhadap bagaimana perusahaan Anti Rayap Malang .com, kami dapat menyimpulkan masalah yang dihadapi oleh perusahaan ini merupakan permasalahan yang umum dihadapai oleh perusahaan yang akan menerapkan manajemen yang terkomputasi dengan baik. Masalah yang dihadapi dapat diselesaikan dengan menemukan alur pola pekerjaan dan menjadikannya baku untuk dapat dijadikan sebuah sistem mandiri yang dapat bekerja dengan baik.

Dengan memanfaatkan beberapa metode seperti halnya normalisasi dan analisis lebih dalam terhadap data sebuah sistem dapat dibangun dengan baik. Dan sistem yang kami maksud sudah kami rumuskan bentuk awalnya pada poin-poin pembahasan berikutnya dalam bab ini.

Setelah kami melakukan serangkaian pengujian sistem ini juga dinilai dapat mempermudah manajerial data yang ada dalam perusahaan. Jika sistem sepenuhnya rampung dapat dimanfaatkan untuk integrasi data yang lebih spesifik dan sangat membantu jalannya perusahaan tersebut. Menurut kami sistem semacam ini juga dapat dimanfaatkan di bidanng usaha lain yang sejenis selama analisis yang dilakukan benar dan data yang dibutuhkan tersedia. Tentu hal ini juga dapat bernilai ekonomis jika kita dapat menyediakan layanan semacam ini.

1. **Normalisasi**
2. **UNF**

Pada bagian ini semua data yang ada di instansi atau perusahaan tersebut dikumpulkan dan dijadikan satu untuk mempermudah proses normalisasi di tahap-tahap selanjutnya. Karena ada proses-proses lanjutan yang nantinya akan membantuk relasi untuk tabel-tabel baru seperti yang sudah dijelaskan sebelumnya.

Tabel 3.1 tabel normalisasi untuk UNF

|  |  |
| --- | --- |
| UNF | { nomor\_proyek, besaran\_proyek, tanggal\_mulai, lama\_pekerjaan, kode\_pemilik, nama\_pemilik, kontak\_pemilik, alamat\_pemilik, kode\_karyawan, nama\_karyawan, alamat\_karyawan, kontak\_karyawan, keahlian, nomor\_survei, lokasi\_survei, tanggal\_survei, deskripsi\_kerusakan, nomor\_inventaris, nama\_inventaris, deskripsi\_inventaris, status\_inventaris, kode\_obat, nama\_obat, banyak\_obat, kandungan\_obat } |

1. **1NF**

Proses selanjutnya adalah *First Normalize Form* atau 1NF. Pada proses ini sekumpulan *field* yang tampil berulang-ulang dikelompokkan menjadi satu dan mengambil *primary key* UNF untuk dijadikan *Foreign key.*

Tabel 3.2 tabel normalisasi untuk 1NF

|  |  |
| --- | --- |
| tabel\_proyekabc | {nomor\_proyek, kode\_pemilik, nama\_pemilik, kontak\_pemilik, alamat\_pemilik, kode\_karyawan, nama\_karyawan, alamat\_karyawan, kontak\_karyawan, keahlian, nomor\_inventaris, nama\_inventaris, deskripsi\_inventaris, status\_inventaris, kode\_obat, nama\_obat, banyak\_obat, kandungan\_obat} |

1. **2NF**

Selanjutnya*Second Normalize form* (2NF), tabel yang sudah dipisahkan tadi dipisahkan lagi berdasarkan data mana yang memiliki ketergantungan terhadap suatu key yang dimiliki. Persis seperti tabel yang ada di bawah.

Tabel 3.3 tabel normalisasi untuk 2NF

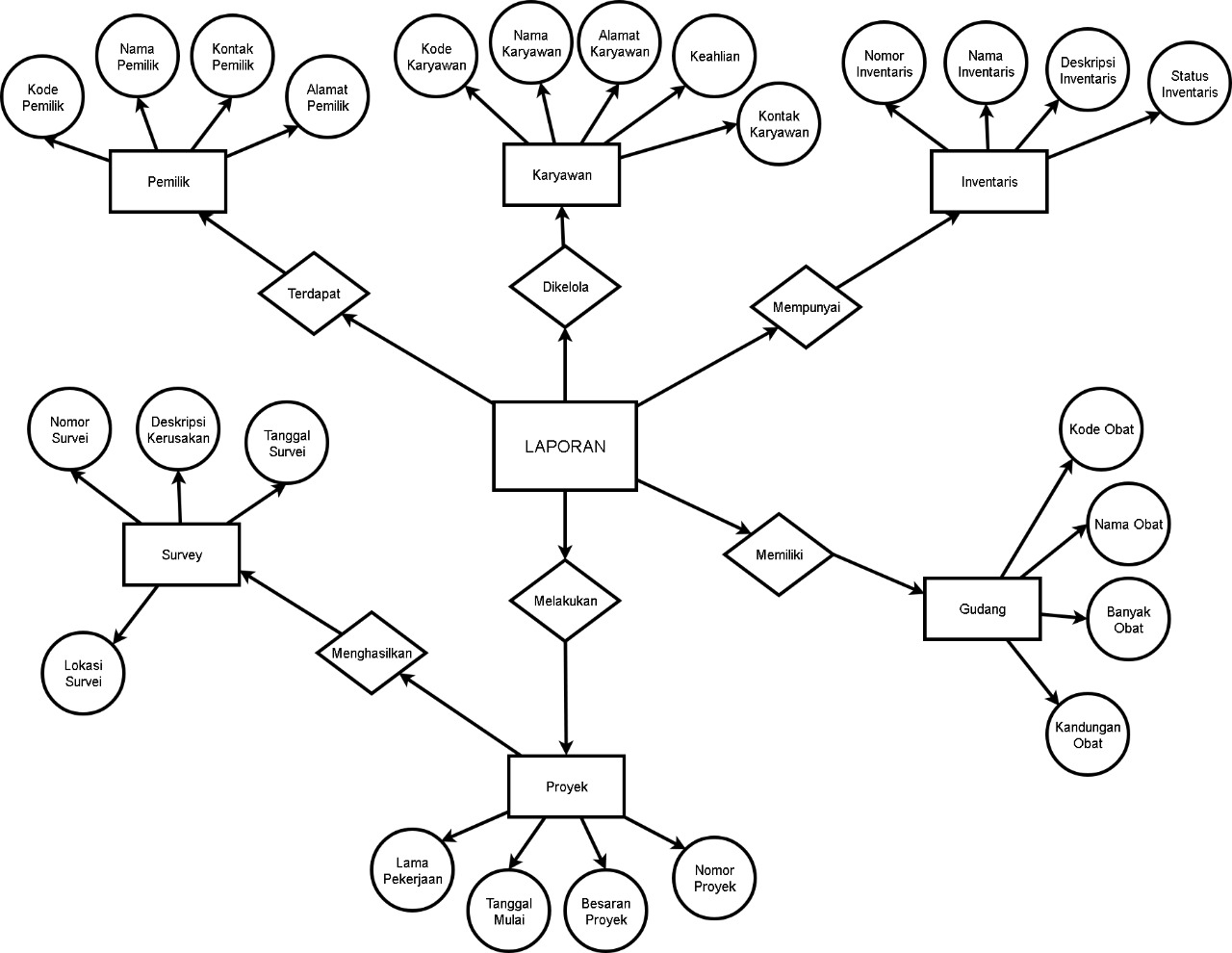
|  |  |
| --- | --- |
| tabel\_laporan2 | {nomor\_proyek,kode\_pemilik,kode\_karyawan,nomor\_inventaris,kode\_obat} |
| tabel\_pemilik2 | {kode\_pemilik,nama\_pemilik,kontak\_pemilik,alamat\_pemilik} |
| tabel\_karyawan2 | {kode\_karyawan,nama\_karyawan,alamat\_karyawan, kontak\_karyawan, keahlian} |
| tabel\_inventaris2 | {nomor\_inventaris,nama\_inventaris,deskripsi\_inventaris,status\_inventaris} |
| tabel\_gudang2 | {kode\_obat,nama\_obat,banyak\_obat,kandungan\_obat} |
| tabel\_proyek2 | {Nomor\_proyek, besaran\_proyek, tanggal\_mulai, lama\_pekerjaan, nomor\_survei, lokasi\_survei, tanggal\_survei, deskripsi\_kerusakan} |

1. **3NF**

Proses terakhir dari rangkaian normalisasi kali ini adalah*Third Normalize Form*(3NF), tentunya pada proses ini data sudah memenuhi syarat pada proses-proses sebelumnya. Pada proses ini kita pastikan sudah tidak terjadi ketergantungan pada atribut *non key* lainnya. Pada tahap ini juga kita dapat menambahkan field-field baru untuk memenuhi tabel seperti tabel di bawah

Tabel 3.4 tabel normalisasi untuk 3NF

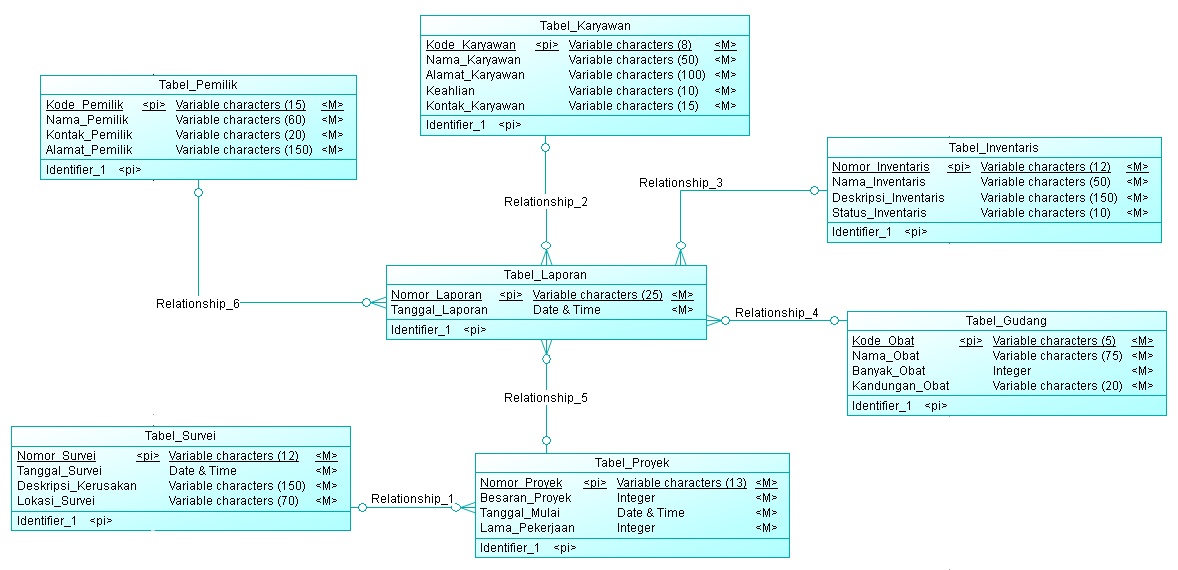
|  |  |
| --- | --- |
| tabel\_laporan | {nomor\_laporan, nomor\_proyek, kode\_pemilik, kode\_karyawan, nomor\_inventaris, kode\_obat, tanggal\_laporan} |
| tabel\_pemilik | {kode\_pemilik, nama\_pemilik, kontak\_pemilik, alamat\_pemilik} |
| tabel\_karyawan | {kode\_karyawan, nama\_karyawan, alamat\_karyawan, kontak\_karyawan, keahlian} |
| tabel\_inventaris | {nomor\_inventaris, nama\_inventaris, deskripsi\_inventaris, status\_inventaris} |
| tabel\_gudang | {kode\_obat, nama\_obat, banyak\_obat, kandungan\_obat} |
| tabel\_proyek | {nomor\_proyek, besaran\_proyek, tanggal\_mulai, lama\_pekerjaan, nomor\_survei} |
| tabel\_survei | {nomor\_survei, lokasi\_survei, tanggal\_survei, deskripsi\_kerusakan} |

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

Gambar 3.1 Gambar ERD dari basis data

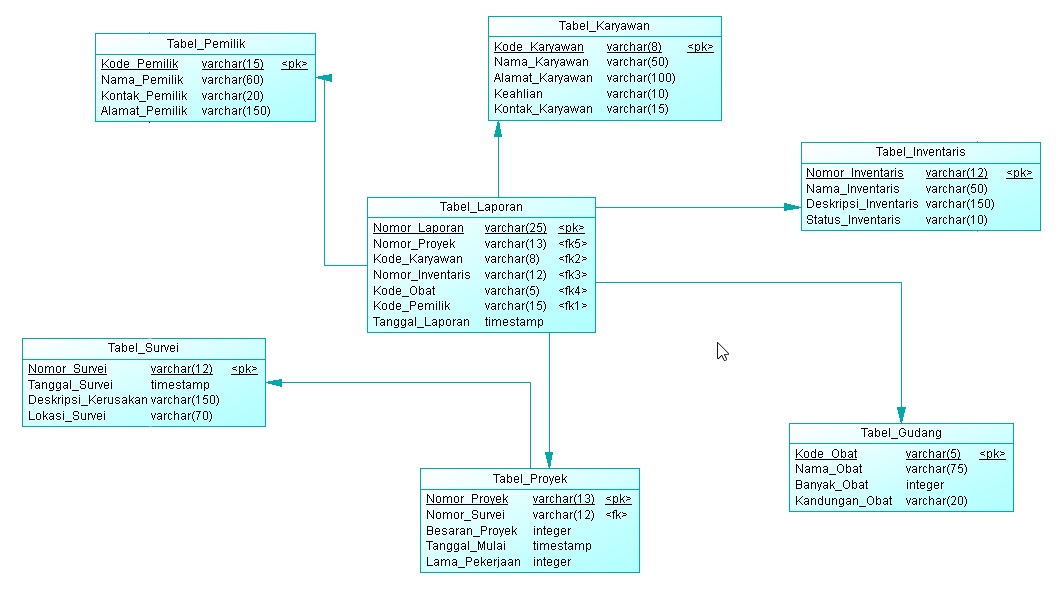
Pada diagram diatas dapat kita amati pola dari bagaimana relasi antar tabel yang ada pada basis data tersebut. Pada dasarnya data tersebut semuanya terhubung pada tabel\_laporan yang merupakan tabel utama yang terhubung dengan tabel lain dan ditambah dengan sebuah tabel lain yaitu tabel\_survei yang terhubung hanya menuju tabel proyek.

Sehingga dari sini kita sudah bisa membayangkan *query* seperti apa yang mungkin akan kita gunakan pada saat basis data tersebut siap digunakan nanti dan juga diagram ini dapat membantu proses pengerjaan pada diagram-diagram lainnya.

1. ***Conceptual Diagram Model* (CDM) & *Physhical Diagram Model* (PDM)**
2. (*Conceptual Diagram Model*) CDM

Gambar 3.2 Gambar CDM dari basis data

Sebuah diagram yang menjadi konsep awal sebelum ditrerapkan dalam bentuk basis datasebelumnya, pada bagian ini kita menentukan beberapa bagian penting dari basis data ini yaitu kardinalitas dari setiap tabelnya. Pada tabel diatas dapat diperhatikan setiap *relationship* yang dimiliki oleh setiap tabel. Biasanya CDM akan dirubah ke bentuk selanjutnya, yaitu PDM yang nanti akan muncul beberapa perubahan signifikan pada PDM

1. (*Physhical Diagram Model*) PDM

Gambar 3.3 Gambar PDM dari basis data

Pada tahap ini, apa yang kami buat pada *diagram* sebelumnya akan dirubah ke dalam bentuk diagram yang sudah siap pakai dan siap diimplementasikan pada basis data yang sesungguhnya. Pada bagian ini jika diamati lebih jauh sebenarnya hampir sama dengan CDM namun pada PDM kardinalitas dan relasi antar tabel lebih jelas dan memang benar-benar mudah untuk mengimplementasikannya dalam bentuk basis data ketika diagram sudah berbentuk seperti in

1. **Tabel**

Di bawah ini adalah beberapa tabel yang menunjukkan *field* dan tipe data yang digunakan pada *field* tersebut.

1. tabel\_laporan

Tabel ini sebenarnya adalah tabel yang tercipta dari relasi antara tabel yang ada untuk dapat menemukan data yang dicari. Dan tabel ini memiliki relasi ke semua tabel yang ada pada basis data ini.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *field* | Tipe data | *Length* | *Key* |
| 1 | nomor\_laporan | varchar | 25 | *Primary Key* |
| 2 | nomor\_proyek | varchar | 13 | *Foreign Key* |
| 3 | kode\_pemilik | varchar | 15 | *Foreign Key* |
| 4 | kode\_karyawan | varchar | 8 | *Foreign Key* |
| 5 | nomor\_inventaris | varchar | 12 | *Foreign Key* |
| 6 | kode\_obat | varchar | 5 | *Foreign Key* |
| 7 | tanggal\_laporan | datetime | - | *Current\_Timestamp* |

Tabel 3.5 tabel field tabel\_laporan

1. tabel\_gudang

Tabel\_gudang adalah sebuah tabel yang berisi data bahan-bahan kimia dari perusahaan terebut. Disebut tabel gudang karena pemilik perusahaan selalu menggunakan instilah obat dan gudang untuk menggambarkan bahan-bahan kimia yang dimiliki (*inventory*).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *field* | Tipe data | *Length* | *Key* |
| 1 | kode\_obat | Varchar | 5 | *Primary Key* |
| 2 | nama\_obat | Varchar | 75 |  |
| 3 | banyak\_obat | Int | 4 |  |
| 4 | kandungan\_obat | varchar | 20 |  |

Tabel 3.6 tabel field tabel\_gudang

1. tabel\_inventaris

Tabel\_inventaris hampir sama dengan tabel\_gudang, namun tabel ini memiliki maksud untuk mengelola barang-barang atau perkakas yang menjadi inventaris dari perusahaan tersebut. Sedikit berbeda dengan tabel\_gudang.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *field* | Tipe data | *Length* | *Key* |
| 1 | nomor\_inventaris | Varchar | 12 | *Primary Key* |
| 2 | nama\_inventaris | Varchar | 50 |  |
| 3 | deskripsi\_inventaris | varchar | 150 |  |
| 4 | status\_inventaris | varchar | 10 |  |

Tabel 3.7 tabel field tabel\_inventaris

1. tabel\_karyawan

Tabel\_karyawan berfungsi untuk mengelola data karyawan yang dimiliki oleh perusahaan agar dapat diolah dengan lebih mudah.

Tabel 3.8 tabel field tabel\_karyawan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *field* | Tipe data | *Length* | *Key* |
| 1 | kode\_karyawan | varchar | 8 | *Primary Key* |
| 2 | nama\_karyawan | varchar | 50 |  |
| 3 | alamat\_karyawan | varchar | 100 |  |
| 4 | Keahlian | varchar | 10 |  |
| 5 | kontak\_karyawan | varchar | 15 |  |

1. tabel\_pemilik

Tabel\_pemilik adalah tabel yang memiliki fungsi untuk mengelola data pemilik atau *owner* dari setiap proyek yang dikerjakan oleh perushaan ini. Setiap pemilik bisa memiliki alamat yang berbeda dengan proyek dan seorang pemilik bisa memberikan beberapa proyek sekaligus.

Tabel 3.9 tabel field tabel\_pemilik

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *field* | Tipe data | *Length* | *Key* |
| 1 | kode\_pemilik | Varchar | 15 | *Primary Key* |
| 2 | nama\_pemilik | Varchar | 60 |  |
| 3 | kontak\_pemilik | Varchar | 20 |  |
| 4 | alamat\_pemilik | varchar | 150 |  |

1. tabel\_proyek

Tabel\_proyek mengelola proyek yang dikerjakan oleh perusahaan ini dan tabel ini memiliki relasi dengan tabel\_survei. Dimana satu proyek pasti selalu dilakukan satu survei.

Tabel 3.10 tabel field tabel\_proyek

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *Field* | Tipe data | *Length* | *Key* |
| 1 | nomor\_proyek | varchar | 13 | *Primary Key* |
| 2 | besaran\_proyek | Int | 11 |  |
| 3 | tanggal\_mulai | datetime | - | *Current\_Timestamp* |
| 4 | lama\_pekerjaan | Int | 3 |  |
| 5 | nomor\_survei | Varchar | 13 | *Foreign Key* |

1. tabel\_survei

Tabel yang terakhir adalah tabel\_survei, tabel ini menampung hasil survei dang menyambungkannya ke sebuah proyek yang tercatat pada tabel\_proyek yang ada.

Tabel 3.11 tabel field tabel\_survei

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | *field* | Tipe data | *Length* | *Key* |
| 1 | kode\_survei | varchar | 12 | *Primary Key* |
| 2 | tanggal\_survei | datetime | - | *Current\_Timestamp* |
| 3 | deskripsi\_kerusakan | varchar | 150 |  |
| 4 | lokasi\_survei | varchar | 70 |  |