# **LAPORAN AKHIR**

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI MANAJEMEN ANTI RAYAP MALANG . COM**



Disusun Oleh :

|  |  |
| --- | --- |
| Panji Iman Baskoro | (171-111-023) |
| Stefanus Salem | (171-111-010) |

**PRODI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA**

**MALANG**

**2018**

# **Kata Pengantar**

Daftar Isi

[LAPORAN AKHIR i](#_Toc518738325)

[Kata Pengantar ii](#_Toc518738326)

[Daftar Isi iii](#_Toc518738327)

[Daftar Tabel v](#_Toc518738328)

[Daftar Gambar vi](#_Toc518738329)

[BAB I 1](#_Toc518738330)

[Pendahuluan 1](#_Toc518738331)

[1.1. Latar Belakang 1](#_Toc518738332)

[1.2. Rumusan Masalah 2](#_Toc518738333)

[1.3. Batasan Masalah 2](#_Toc518738334)

[1.4. Tujuan 3](#_Toc518738335)

[1.5. Manfaat 3](#_Toc518738336)

[BAB II 4](#_Toc518738337)

[Landasan Teori 4](#_Toc518738338)

[2.1 Apache HTTP *Server* 4](#_Toc518738339)

[2.2 *Database* 4](#_Toc518738340)

[2.3 MySQL 5](#_Toc518738341)

[2.4 Normalisasi *Database* 5](#_Toc518738342)

[a. 1NF (*First Normal Form*) 5](#_Toc518738343)

[b. 2NF (*Second Normal Form*) 5](#_Toc518738344)

[c. 3NF (*Third Normal Form*) 5](#_Toc518738345)

[2.5 PHP 6](#_Toc518738346)

[2.6 PHPMyadmin 6](#_Toc518738347)

[2.7 Table 7](#_Toc518738348)

[2.8 XAMPP 7](#_Toc518738349)

[2.9 ERD (*Entity Relationship Diagram*) 7](#_Toc518738350)

[2.10 CDM (*Conseptual Data Model*) 10](#_Toc518738351)

[2.11 PDM (*Physhical Data Model*) 10](#_Toc518738352)

[BAB III 11](#_Toc518738353)

[Analisa dan Desain 11](#_Toc518738354)

[3.1 Analisis Permasalahan 11](#_Toc518738355)

[3.2 Normalisasi 11](#_Toc518738356)

[a. UNF 11](#_Toc518738357)

[b. 1NF 12](#_Toc518738358)

[c. 2NF 12](#_Toc518738359)

[d. 3NF 13](#_Toc518738360)

[3.3 *Entity Relationship Diagram* (ERD) 14](#_Toc518738361)

[3.4 *Conceptual Diagram Model* (CDM) & *Physhical Diagram Model* (PDM) 15](#_Toc518738362)

[3.5 Tabel 16](#_Toc518738363)

[BAB IV 20](#_Toc518738364)

[Implementasi 20](#_Toc518738365)

[4.1 Baris Perintah SQL 20](#_Toc518738366)

[BAB V 33](#_Toc518738367)

[Kesimpulan dan Saran 33](#_Toc518738368)

[5.1. Kesimpulan 33](#_Toc518738369)

[5.2. Saran dan Kritik 33](#_Toc518738370)

**Daftar Tabel**

Tabel 2.1 Tabel symbol-simbol dalam ERD8

Tabel 3.1 tabel normalisasi untuk UNF **12**

Tabel 3.2 tabel normalisasi untuk 1NF 12

Tabel 3.3 tabel normalisasi untuk 2NF 13

Tabel 3.4 tabel normalisasi untuk 3NF 13

Tabel 3.5 tabel field tabel\_laporan 17

Tabel 3.6 tabel field tabel\_gudang 17

Tabel 3.7 tabel field tabel\_inventaris 18

Tabel 3.8 tabel field tabel\_karyawan 18

Tabel 3.9 tabel field tabel\_pemilik 18

Tabel 3.10 tabel field tabel\_proyek 19

Tabel 3.11 tabel field tabel\_survei 19

**Daftar Gambar**

Gambar 3.1 Gambar ERD dari basis data 14

Gambar 3.2 Gambar CDM dari basis data 15

Gambar 3.3 Gambar PDM dari basis data 16

Gambar 4.1 membuat database baru dan hasil database yang dibuat 20

Gambar 4.2 membuat tabel\_survei beserta hasilnya 21

Gambar 4.3 membuat tabel\_proyek beserta hasilnya 21

Gambar 4.4 membuat tabel\_gudang beserta hasilnya 22

Gambar 4.5 membuat tabel\_inventaris beserta hasilnya 22

Gambar 4.6 membuat tabel\_pemilik beserta hasilnya 23

Gambar 4.7 membuat tabel\_karyawan beserta hasilnya 23

Gambar 4.8 membuat tabel\_laporan beserta hasilnya 24

Gambar 4.9 memasukkan data ke tabel\_gudang beserta hasilnya 25

Gambar 4.10 memasukkan data ke tabel\_karyawan beserta hasilnya 25

Gambar 4.11 memasukkan data ke tabel\_inventaris beserta hasilnya 26

Gambar 4.12 memasukkan data ke tabel\_pemilik beserta hasilnya 26

Gambar 4.13 memasukkan data ke tabel\_proyek beserta hasil 27

Gambar 4.14 memasukkan data ke tabel\_proyek beserta hasilnya 28

Gambar 4.15 memasukkan data ke tabel\_laporan beserta hasilnya 29

Gambar 4.16 mengambil data dari tabel dengan metode join 30

Gambar 4.17 membuat view dari sebuah query join 31

Gambar 4.18 query join untuk beberapa tabel sekaligus 32

# **BAB I**

# **Pendahuluan**

1. **Latar Belakang**

Perkembangan teknologi belakangan ini sangatlah luar biasa. Terbukti dari munculnya beraneka ragam inovasi diberbagai bidang yang ditopang oleh teknologi tertentu yang sangat membantu aktifitas manusia sehari-hari. Perkembangan ini juga didorong oleh begitu cepatnya kita dapat memperoleh informasi melalui internet, di Indonesia sendiri pada tahun 2017 ada ± 143.3 juta pengguna internet (Survei APJII tahun 2017 ~ rilis 2018) yang juga turut berkontribusi terhadap arus pertukaran data dan kemajuan teknologi yang ada.

Dengan begitu tingginya penggunaan teknologi khususnya internet dalam kehidupan sehari-hari tidak dapat dipungkiri lagi. Internet telah menjadi bagian dari kehipan kita, internet juga telah berubah bukan lagi sekedar tempat mencari dan berbagi informasi tapi juga menjadi tempat untuk memenuhi kebutuhan hidup kita sehari-hari. Hal tersebut juga membuat banyak pemilik usaha maupun jasa yang beramai-ramai menyediakan layanan mereka dalam versi daring sebagai bentuk jawaban dari permintaan pasar yang mulai bergeser ke pemasaran berbentuk *online* atau melalui internet.

Cara berbisnis yang telah berubah mengarah ke digitalisasi memiliki tantangan tersendiri khususnya bagi para pelaku bisnis. Pelaku bisnis saat ini dituntut untuk melayani pelanggan dengan lebih cepat dan tepat. Hal tersebut harus diiringi dengan manajerial yang tidak kalah cepatnya dalam merespon segala macam perubahan yang terjadi di dalam perusahaan tersebut. Manajerial yang cepat harus diiringi dengan sebuah sistem yang memang khusus ditujukan untuk membantu berjalannya proses manajerial dengan baik. Tidak hanya perusahaan berskala besar namun juga usaha kecil masyarakat juga perlu memiliki Sistem serupa yang telah disesuaikan. Kembali lagi karena persaingan yang ketat harus diiringi dengan analisis yang tepat pula oleh para pelaku bisnis. Sehingga kita sepakat jika kehadiran sebuah Sistem Informasi Manajemen memang benar-benar dibutuhkan oleh berbagai jenis bisnis entah itu bisnis kecil maupun korporat besar.

Oleh karena itu, kami berupaya untuk membuat sebuah Sistem Informasi Manajemen yang khusus ditujukan untuk pelaku bisnis kecil di Indonesia. Hal ini perlu dilakukan agar setiap usaha kecil di Indonesia juga dapat bersaing dengan kemampuan manajerial yang sebanding bahkan dengan perusahaan berskala menengeh keatas sekalipun. Atas dasar itu kami memilih judul “Membangun *Database* Sistem Informasi Manajemen untuk Anti Rayap Malang .com” untuk dijadikan judul tugas kami dan bahan analisis kedepannya.

1. **Rumusan Masalah**

Berdasarkan uraian diatas maka dapat dirumuskan permasalahan penelitian ini sebagai berikut :

1. Bagaimana cara membangun sebuah Sistem Informasi Manajemen beserta dengan *database* yang sesuai untuk bisnis kecil?
2. Bagaimana rancangan sistem dan *database* yang akan dibangun?
3. Bagaimana Sistem Informasi berkontribusi terhadap jalannya perusahaan kecil?
4. **Batasan Masalah**

Agar proposal ini tidak menyimpang dari tujuan yang semula direncanakan maka kami menetapkan batasan-batasan sebagai berikut :

1. Mengidentifikasi cara membangun Sistem Informasi Manajemen untuk bisnis kecil
2. Membuat rancangan sistem dan *database* untuk Sistem Informasi Manajemen Anti Rayap Malang.com
3. Analisis kontribuisi Sistem Informasi Manajemen terhadap perkembangan perusahaan
4. Pengimplementasian Sistem Informasi manajemen secara umum
5. **Tujuan**

Membangun sebuah *database* serta Sistem Informasi Manajemen untuk bisnis menengah kebawah yang disesuaikan dengan kebutuhan dari perusahaan yang bersangkutan serta melakukan analisis sederhana terhadap kontribusi Sistem Manajemen yang berjalan nantinya.

1. **Manfaat**

Manfaat Dari Penelitian yang kami buat adalah :

1. Bagi Penulis
   1. Penulis Dapat Mempelajari Bagaimana cara Membuat *database* sistem informasi manajemen yang digunakan dalam sebuah perusahaan.
   2. Dapat mngumpulkan data analisis tentang *database* Sistem Informasi Manajemen khususnya bagi perusahaan menengah ke bawah.
   3. Dapat melakukan analisis singkat mengenai pemanfaatan Sistem Informasi Manajemen di lingkungan perushaan tersebut.
2. Bagi Perusahaan
3. Dapat mengatasi permasalahan manajemen dari sebuah perusahaan.
4. Dapat membangun sebuah sistem yang bersahabat terhadap lingkungan perusahaaan atau usaha kecil menengah.
5. Sistem yang akan kami bangun tentu akan bermanfaat bagi perusahaan yang telah memberi kami izin untuk melakukan analisis di lingkungan perusahaan tersebut .

# **BAB II**

# **Landasan Teori**

1. **Apache HTTP *Server***

*Apache HTTP server* adalah proyek opensource dibawah naungan *Apache Software Foundation* salah satu komunitas opensource terbesar di dunia. Apache HTTP *server* (httpd) adalah salah satu aplikasi *web server* yang paling banyak digunakan di dunia. Meskipun memiliki banyak saingan *web server* seperti IIS dari Windows dan *Litespeed* salah satu aplikasi *web server* tercepat saat ini, masih banyak teknisi yang enggan migrasi karena dokumentasi untuk httpd yang lebih rapi dan solid karena dukungan yang cukup besar untuk proyek *opensource* ini. Selain itu konfigurasi *server* yang lebih mudah turut menjadi alasan kenapa httpd dinobatkan sebagai aplikasi *web server* yang paling banyak digunakan oleh pengembang dan penyedia layanan *web server* di seluruh dunia saat ini dibanding applikasi lain. (https://httpd.apache.org/ABOUT\_APACHE.html)

1. ***Database***

*Database* atau basis data adalah kumpulan data yang sidimpan secara sistimatis dan terkomputerisasi dengan baik agar bisa dimanfaatakan kapanpun dengan lebih mudah. *Database* semakin sering digunakan karena dapat mengurangi penggunaan kertas untuk pengumpulan data khususnya pada perkantoran. Ditambah dengan proses pengolahan data yang lebih membuat penggunaan databas yang tepat dan efisien adalah solusi praktis dalam manajemen data dalam suatu perusahaan dan lebih hemat biaya operasional yang biasa dikeluarkan dan bisa dengan mudah dikembangkan lagi.(https://www.termasmedia.com/lainnya/software/69-pengertian-*Database*.html)

1. **MySQL**

Dalam penggunaannya *Database* memerlukan bantuan sebuah aplikasi manajemen sistem *Database* yang dapat membantu kita untuk menambahkan, menghapus, mengambil, dan mengubah data yang ada pada *Database*. Aplikasi ini sering disebut dengan DBMS (*Database* *Management* *System*), salah satu contohnya adalah MySQL yang merupakan salah satu aplikasi manajemen sistem yang paling sering digunakan oleh pengembang. Namun aplikasi ini memiliki dukungan yang rendah terhadap sebuah error yang terjadi di sistemnya karena aplikasi ini bersifat gratis. Namun dalam penggunaan sistem yang masih berskala kecil penggunaan MySQL sangatlah efektif dan dapat diandalkan selama penggunaan syntax-nya diperhatikan dengan baik, karena perbedaan penggunaan syntax pada mysql dapat menghasilkan kecepatan pengolahan data yang berbeda dan akurasi yang berbeda pula. (https://www.termasmedia.com/lainnya/software/69-pengertian-*Database*.html)

1. **Normalisasi *Database***

Sebuah proses pengolahan data mentah yang belum diaplikasikan ke *Database* yang bertujuan untuk menghilangkan anomali atau keanehan pada data. Hal ini bertujuan agar data dapat diolah dengan lebih mudah selain itu juga bertujuan untuk mempermudah proses pembangunan *Database* nantinya. Metode normalisasi terbagi menajdi beberapa tahap yaitu :

* 1. 1NF (*First Normal Form*)

Pada tahapan normalisasi ini yang dilakukan adalah menghilangkan data-data yang berulang – ulang kemunculannya di dalam *Database* dan memisahkannya kedalam tabel lain.

* 1. 2NF (*Second Normal Form*)

Setelah form memenuhi syarat yang ada pada 1NF, bisa kita lanjutkan proses ke 2NF yaitu proses memindahkan data pada tabel yang berbeda dan menghilangkan data yang saling ketergantungan dan memindahkannya ke tabel baru.

* 1. 3NF (*Third Normal Form*)

Setelah selesai melakukan proses normalisasi pada tahap-tahap sebelumnya, yang selanjutnya harus dilakukan adalah menghapus kolom- kolom pada tabel yang tidak tergantung pada *primary key* yang dimiliki dan harus memenuhi syarat-syarat normal form yang ada pada normal form yang ada sebelumnya. (https://rifalfahrudin.wordpress.com/2015/09/27/normalisasi-*Database*-1nf-2nf-3nf/)

1. **PHP**

PHP (*PHP Hyper Preprocessor*) adalah sebuah bahasa pemprograman yang awalnya dikembangkan untuk mengatasi kesulitan para pengembang untuk mengkonfigurasi *form* pada halaman HTML. Kini PHP perkembang menajdi salah satu bahasa pemprograman paling digunakan di dunia. Perusahaan besar seperti Facebook juga menggunakan bahasa pemprograman ini. Hal lain yang istimewa adalah *learning pattern* atau pola pembelajaran PHP yang cenderung lebih landai dan mudah dipelajari oleh pemula. Di Indonesia sendiri PHP memiliki banyak komunitas mulai dari tingkat daerah hingga tingkat nasional. (https://id.wikipedia.org/wiki/PHP)

1. **PHPMyadmin**

PHPMyadmin adalah aplikasi web untuk mengelola *Database* MySQL dan *Database* MariaDB dengan lebih mudah melalui antarmuka (*interface*) grafis. Aplikasi web ini ditulis menggunakan bahasa pemrograman PHP. Sebagaimana aplikasi-aplikasi lain untuk lingkungan web (aplikasi yang dibuka atau dijalankan menggunakan *browser*), phpMyAdmin juga mengandung unsur HTML/XHTML, CSS dan juga kode JavaScript. Aplikasi web ini ditujukan untuk memudahkan pengelolaan basis data MySQL dan MariaDB dengan penyajian antarmuka web yang lengkap dan menarik. (https://www.termasmedia.com/database/phpmyadmin/131-mengenal-phpmyadmin-dan-kegunaannya.html)

1. **Table**

*Table* adalah adalah daftar yang berisi sejumlah data-data informasi atau fakta yang biasanya berupa kata-kata maupun bilangan yang tersusun dengan urutan kolom dan baris, biasanya tabel dalam basis data diartikan sebagai kumpulan dari beberapa *record* dan *fields* . Fungsi Tabel untuk mendukung penulis menyampaikan ide atau gagasan sehingga dapat memengaruhi dan meyakinkan pembaca selain itu tampilan data pada table juga cukup menarik untuk diamati dan mempermudah pekerjaan. (<https://bundaliainsidi.blogspot.com/2015/10/pengertian-field-record-table-file-data.html>)

1. **XAMPP**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP *Server*, MySQL*Database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mengunduh langsung dari web resminya. (https://id.wikipedia.org/wiki/XAMPP)

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

Diagram Hubungan Entitas atau entity relationship diagram merupakan model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpan. Model data sendiri merupakan sekumpulan cara, peralatan untuk mendeskripsikan data-data yang hubungannya satu sama lain, semantiknya, serta batasan konsistensi. Model data terdiri dari model hubungan entitas dan model relasional. Diagram hubungan entitas ditemukan oleh Peter Chen dalam buku *Entity Relational Model-Toward a Unified of Data*. Chen mencoba merumuskan dasar-dasar model dan setelah itu dikembangkan dan dimodifikai oleh Chen dan banyak pakar lainnya. Pada saat itu diagram hubungan entitas dibuat sebagai bagian dari

perangkat lunak yang juga merupakan modifikasi khusus, karena tidak ada bentuk tunggal dan standar dari diagram hubungan entitas. (https://id.wikipedia.org/wiki/Diagram\_hubungan\_entitas, <https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning>)

Tabel 2.1 Tabel simbol – simbol dalam ERD

|  |  |  |
| --- | --- | --- |
| Simbol | Nama Simbol | Deskripsi Simbol |
| *Entitiy* | | |
| ERD Symbols and Meaning - Entity | *Entity* | Simbol *entity* |
| ERD Symbols and Meaning - Weak Entity | *Weak Entity* | *Weak entity* adalah sebuah entitas yang keberadaannya sangat bergantung kepada entitas lain. |
| Atribut | | |
| ERD Symbols and Meaning - Attribute | *Atribut* | Simbol untuk atribut dari *entity* |
| ERD Symbols and Meaning - Key attribute | *Key attribute* | Simbol untuk atribut utama atau bisa disebut *primary key* |
| ERD Symbols and Meaning - Multivalue attribute | *Multivalued attribute* | Simbol untuk atribut yang memiliki *value* lebih dari satu |
| ERD Symbols and Meaning - Derived attribute | *Derived attribute* | Atribut tidak tetap yang dihasilkan dari dua atribut lain yang berbeda |
| ERD Symbols and Meaning - Relationship | *Strong relationship* | Relasi antar 2 entity yang bersifat *independent* |
| ERD Symbols and Meaning - Identifying Relationship | *Weak relationship* | Sebuah relasi yang terbangun dari *weak entity* yang tergantung pada *weak entity* sebagai *child entitiy* |

|  |  |
| --- | --- |
| Simbol | Makna |
| Relasi kardinalitas | |
| ERD symbol -  Entity relationship | *Zero or One* |
| ERD symbol -  Entity relationship | *One or More* |
| ERD symbol -  Entity relationship | *One and only One* |
| ERD symbol -  Entity relationship | *Zero or More* |

1. **CDM (*Conseptual Data Model*)**

CDM (*Conseptual* Data Model) dipakai untuk menggambarkan secara detail struktur basis data dalam bentuk logik. Struktur ini independen terhadap semua *software* maupun struktur data *storage* tertentu yang digunakan dalam aplikasi ini. CDM terdiri dari objek yang tidak diimplementasikan secara langsung kedalam basis data yang sesungguhnya. (https://agusandi.wordpress.com/2010/03/15/apa-itu-cdm-fan-pdm/)

1. **PDM (*Physhical Data Model*)**

PDM merupakan gambaran secara detail basis data dalam bentuk fisik. Penggambaran rancangan PDM memperlihatkan struktur penyimpanan data yang benar pada basis data yang digunakan sesungguhnya. (https://agusandi.wordpress.com/2010/03/15/apa-itu-cdm-fan-pdm/)

# **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

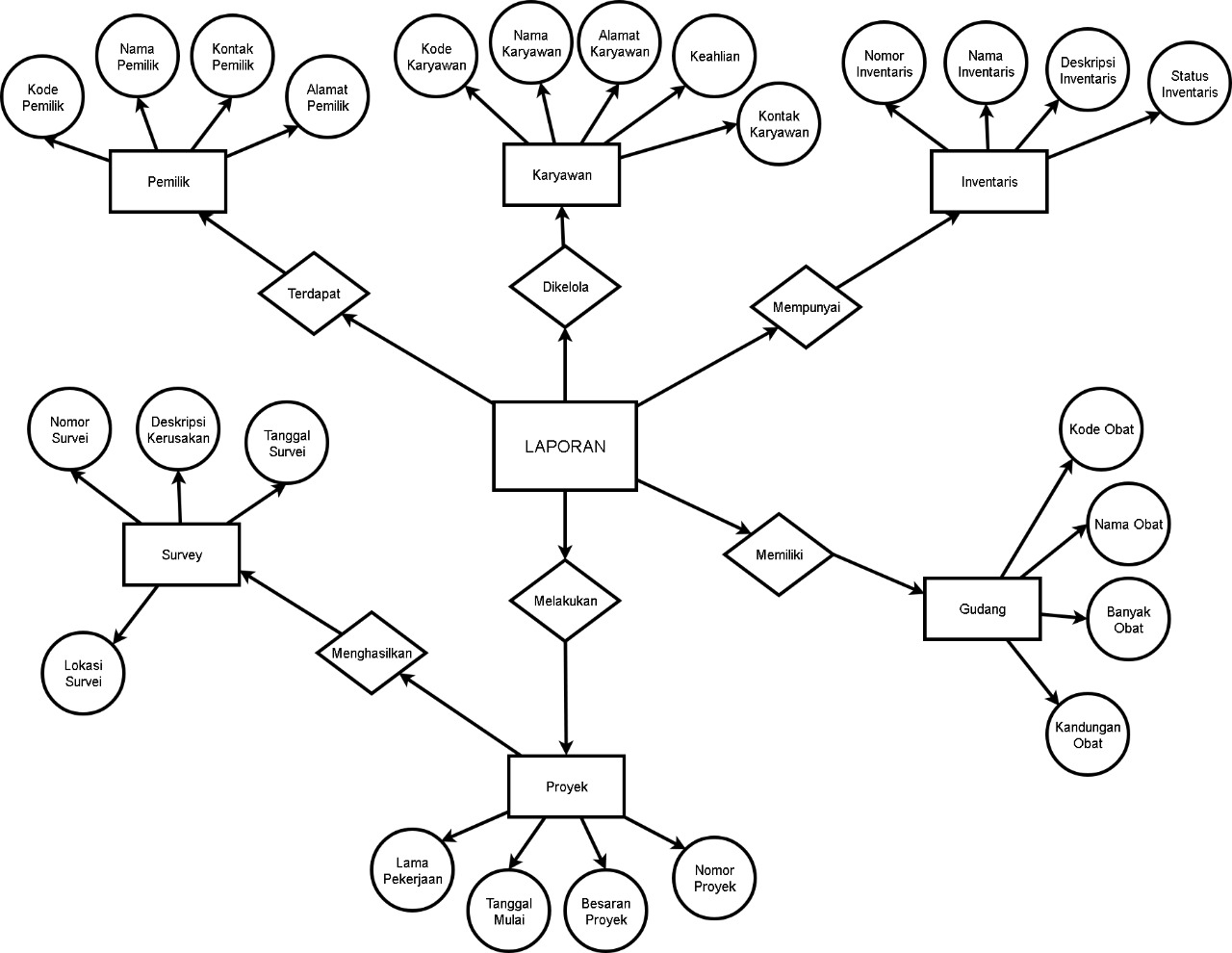
|  |  |
| --- | --- |
| tabel\_laporan2 | {nomor\_proyek,kode\_pemilik,kode\_karyawan,nomor\_inventaris,kode\_obat} |
| tabel\_pemilik2 | {kode\_pemilik,nama\_pemilik,kontak\_pemilik,alamat\_pemilik} |
| tabel\_karyawan2 | {kode\_karyawan,nama\_karyawan,alamat\_karyawan, kontak\_karyawan, keahlian} |
| 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

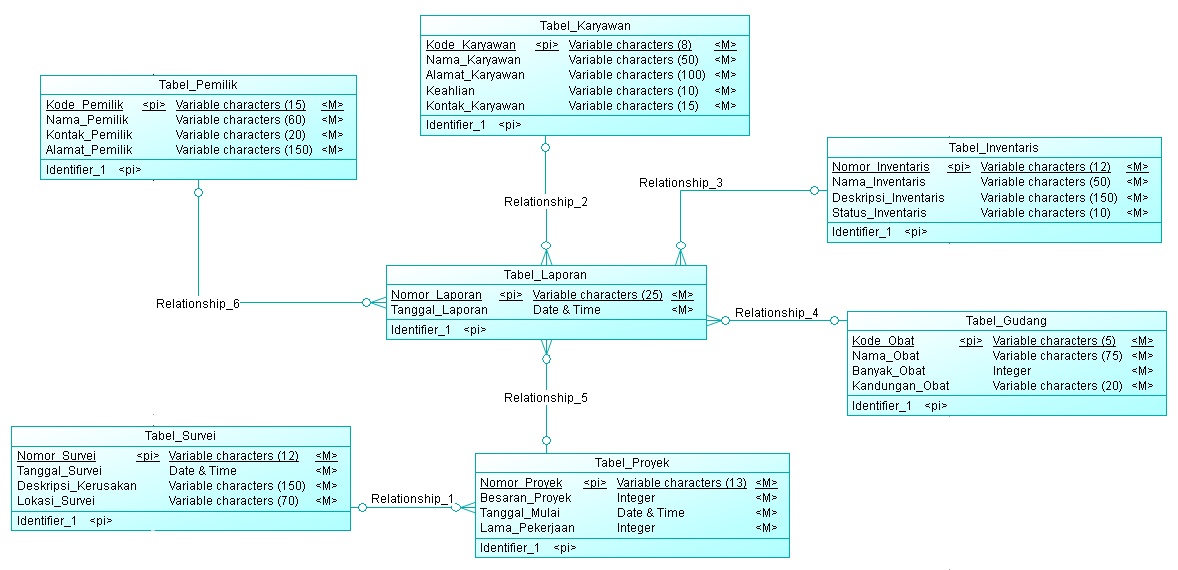
|  |  |
| --- | --- |
| tabel\_laporan | {nomor\_laporan, nomor\_proyek, kode\_pemilik, kode\_karyawan, nomor\_inventaris, kode\_obat, tanggal\_laporan} |
| tabel\_pemilik | {kode\_pemilik, nama\_pemilik, kontak\_pemilik, alamat\_pemilik} |
| tabel\_karyawan | {kode\_karyawan, nama\_karyawan, alamat\_karyawan, 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

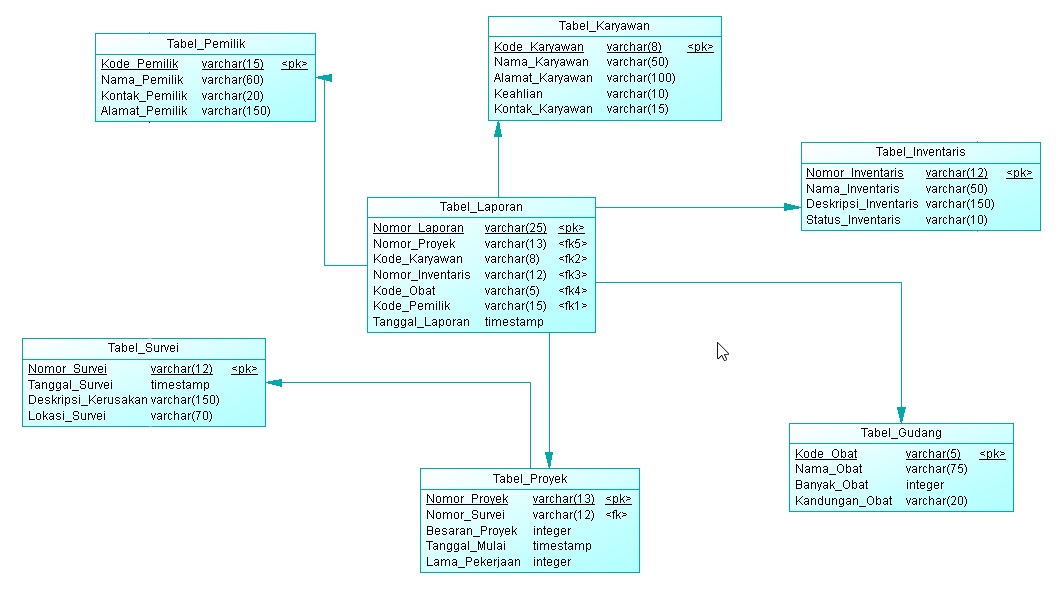
Pada diagram diatas dapat kita amati pola dari bagaimana relasi antar tabel yang ada pada basis data tersebut. Pada dasarnya data tersebut semuanya terhubung pada tabel\_laporan yang merupakan tabel utama yang terhubung dengan tabel lain dan ditambah dengan sebuah tabel lain yaitu tabel\_survei yang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

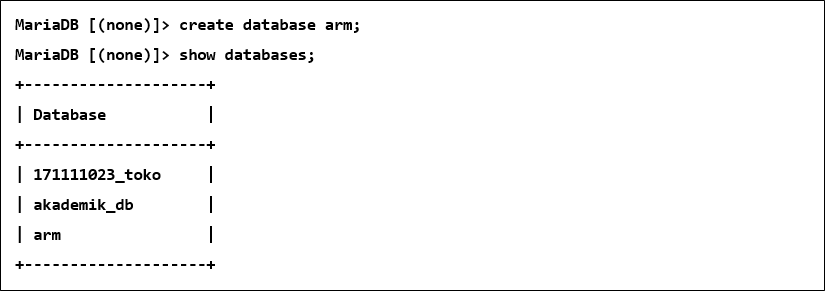
Tabel 3.11 tabel field tabel\_survei

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |  |

# **BAB IV**

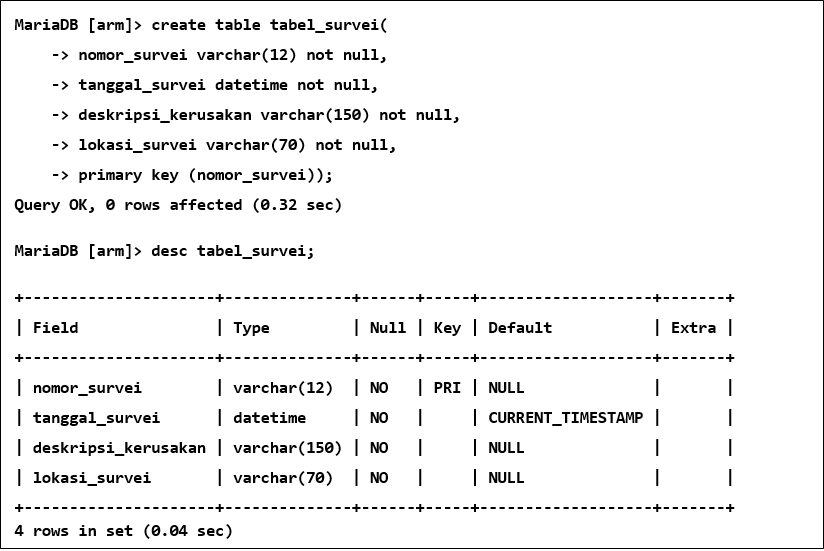
# **Implementasi**

1. **Baris Perintah SQL**
2. Membuat *Database*

Perintah pertama yang kami eksekusi adalah *query* untuk membuat sebuah *database* baru seperti gambar 4.1.

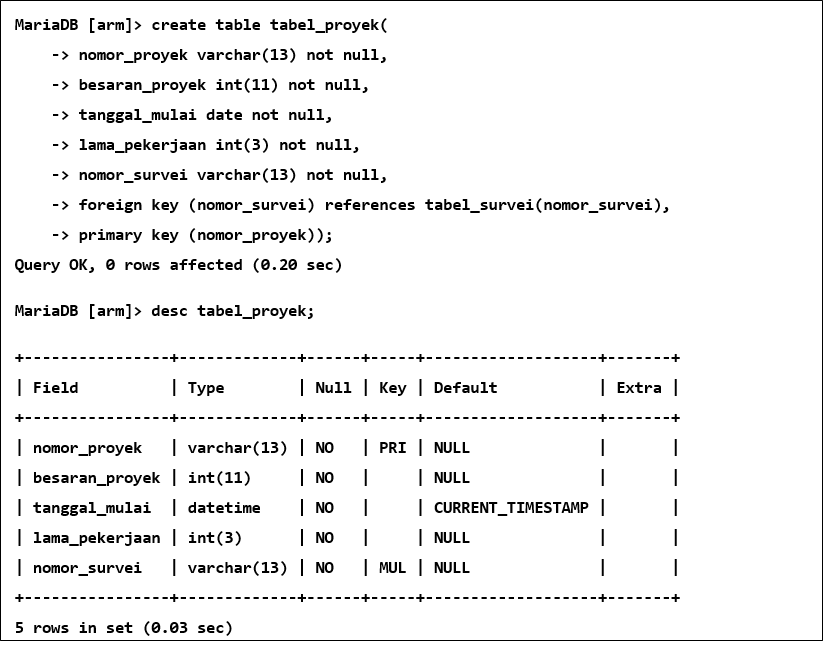
Gambar 4.1 membuat *database* baru dan hasil *database* yang dibuat

1. Membuat tabel\_survei

**Selanjutnya *query* untuk membuat tabel\_survei pada gambar 4.2.

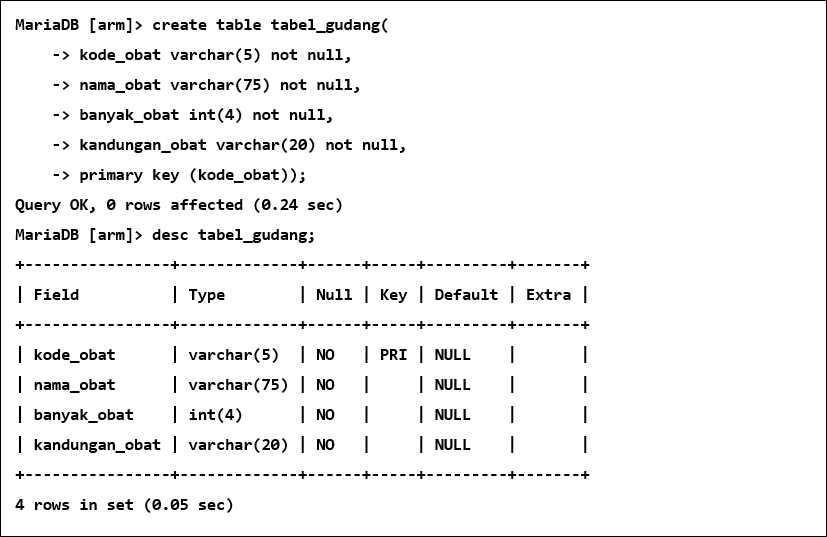
Gambar 4.2 membuat tabel\_survei beserta hasilnya

1. Membuat tabel\_proyek

Pada gambar 4.3 kami membuat tabel\_proyek.

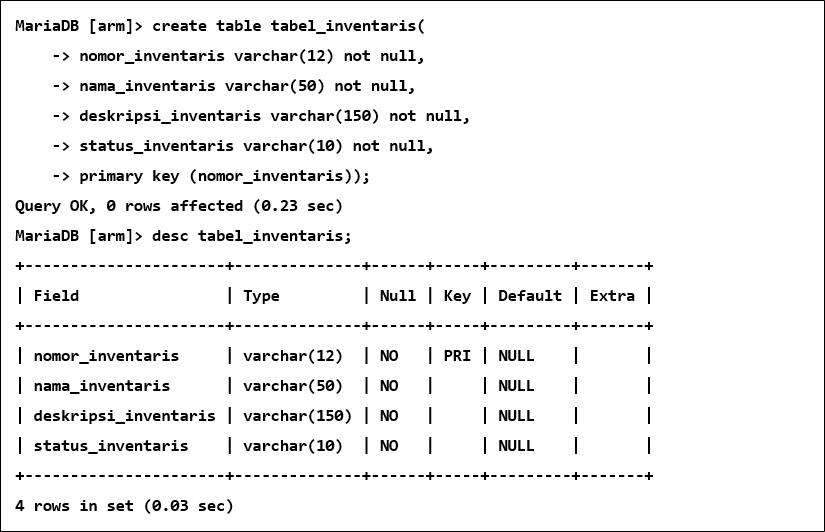
Gambar 4.3 membuat tabel\_proyek beserta hasilnya

1. Membuat tabel\_gudang

Selanjutnya kami membuat tabel\_gudang seperti di bawah.

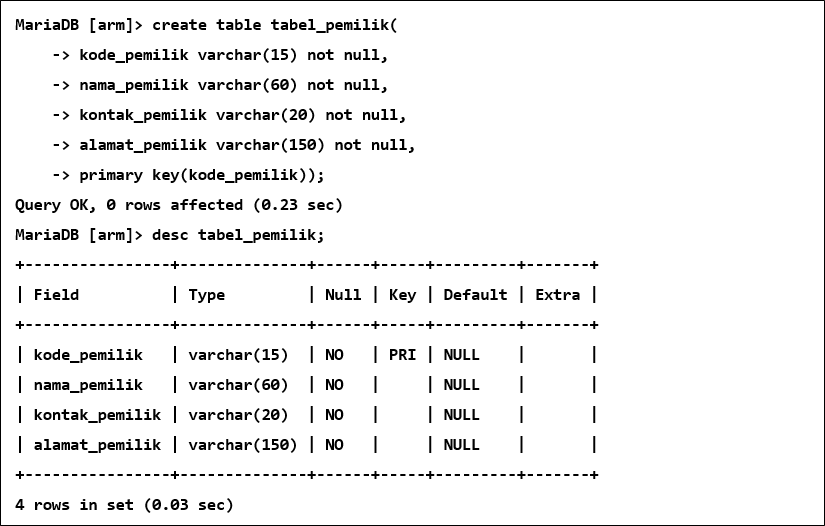
Gambar 4.4 membuat tabel\_gudang beserta hasilnya

1. Membuat tabel\_inventaris

Kemudian di gambar 4.5 kami membuat tabel\_inventaris.

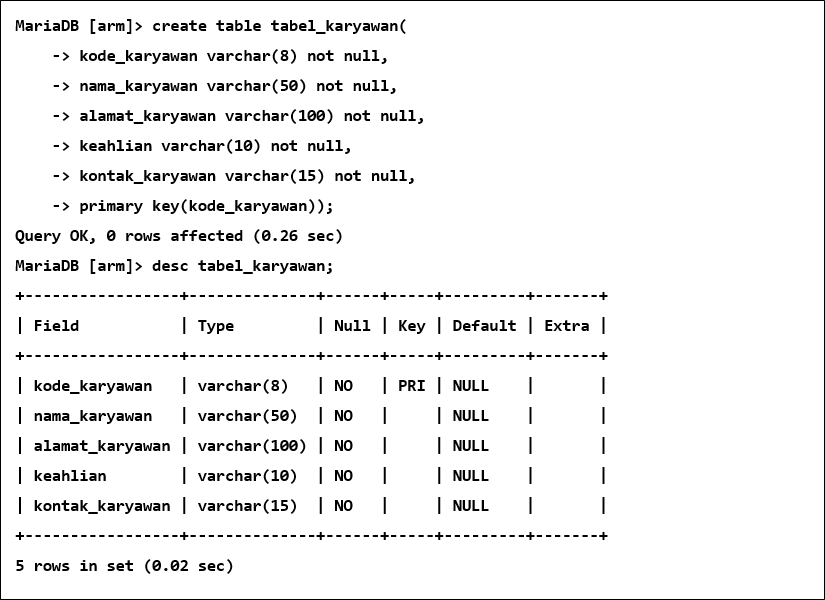
Gambar 4.5 membuat tabel\_inventaris beserta hasilnya

1. Membuat tabel\_pemilik

 Selanjutnya kami membuat tabel baru bernama tabel\_pemilik.

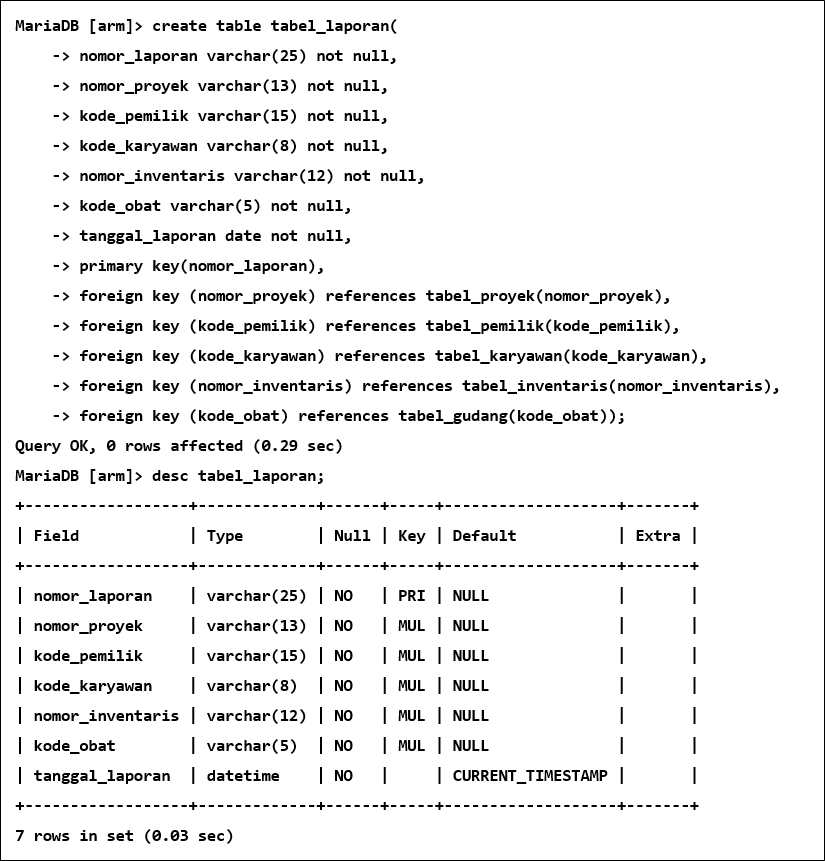
Gambar 4.6 membuat tabel\_pemilik beserta hasilnya

1. Membuat tabel\_karyawan

Kemudian kami membuat tabel baru bernama tabel\_karyawan dengan *query* seperti di bawah.

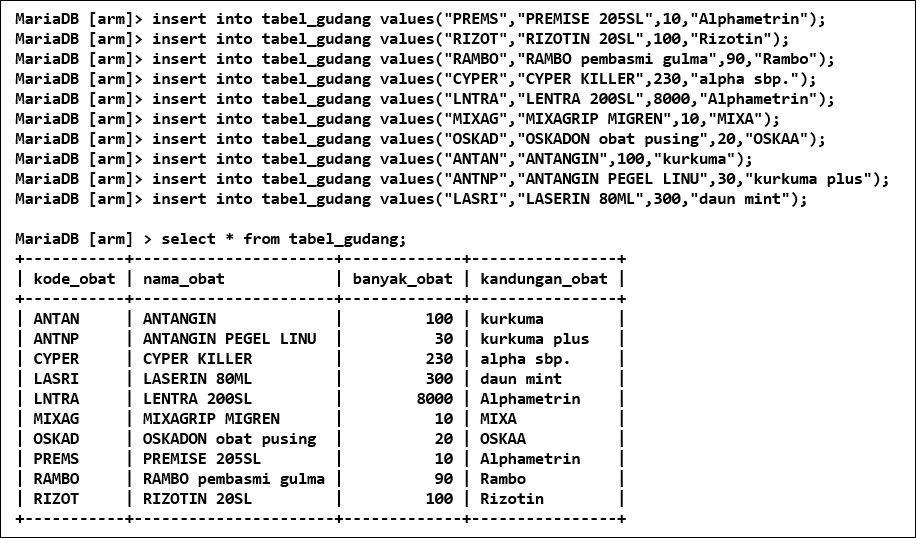
Gambar 4.7 membuat tabel\_karyawan beserta hasilnya

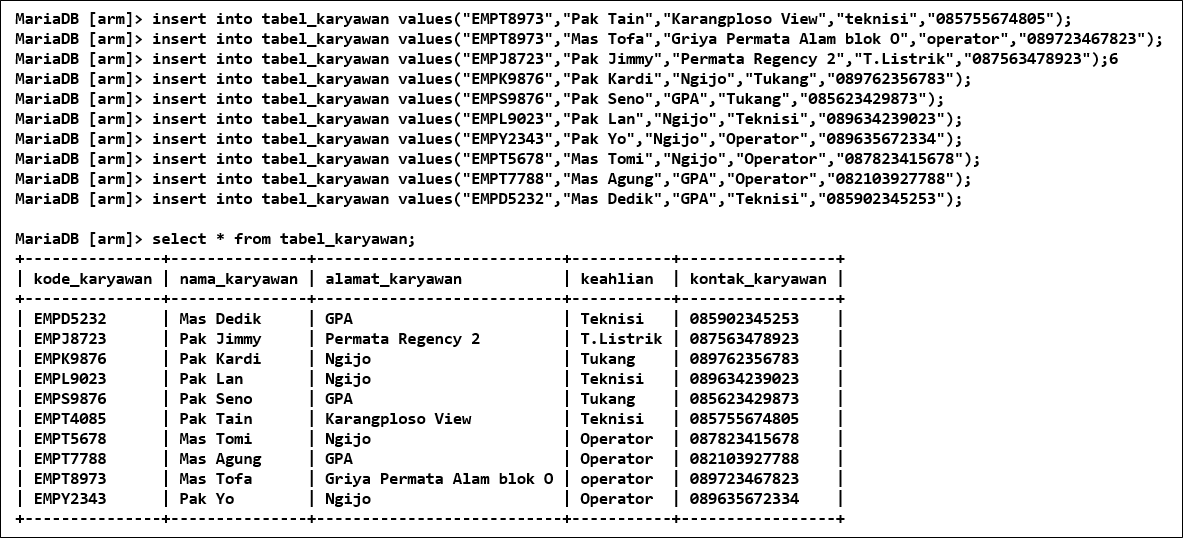
1. Membuat tabel\_laporan

Tabel terakhir yang kami buat adalah tabel\_laporan yang merupakan tabel penghubung relasi antar beberapa tabel lainnya.

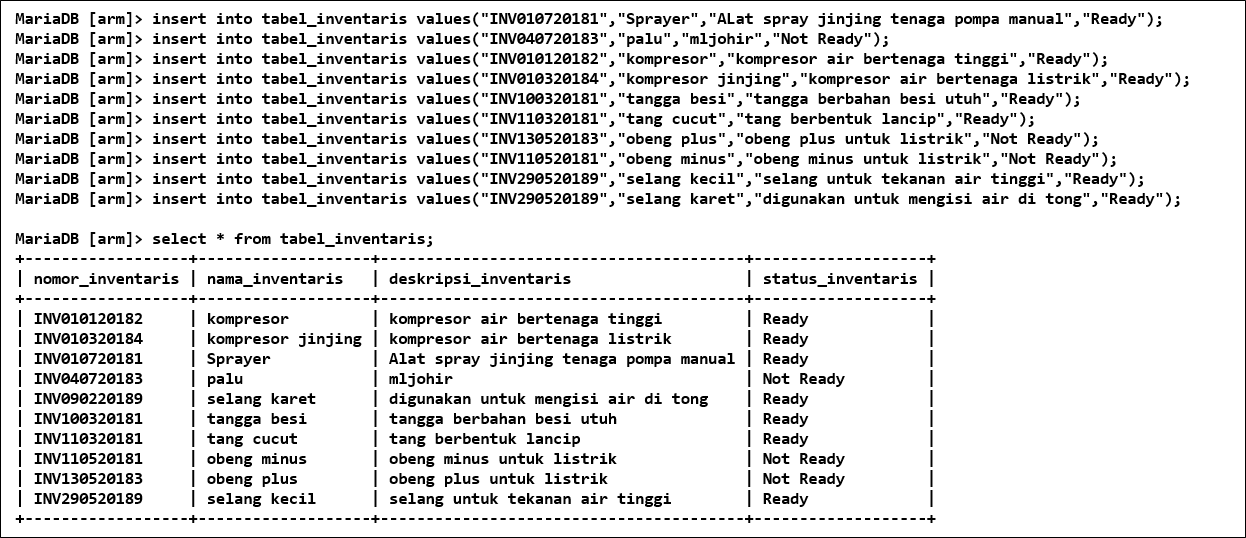
Gambar 4.8 membuat tabel\_laporan beserta hasilnya

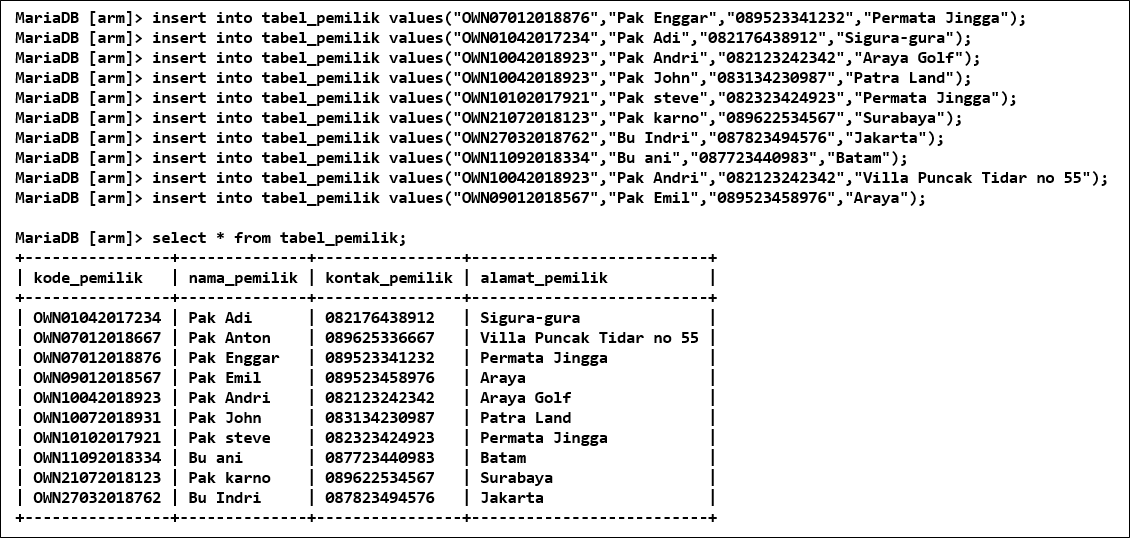
1. *Query* untuk memasukkan data yang akan digunakan

Sebelum kami membuat program sederhana untuk *database* tersebut kami memasukkan beberapa data pada tabel menggunakan perintah seperti di bawah.

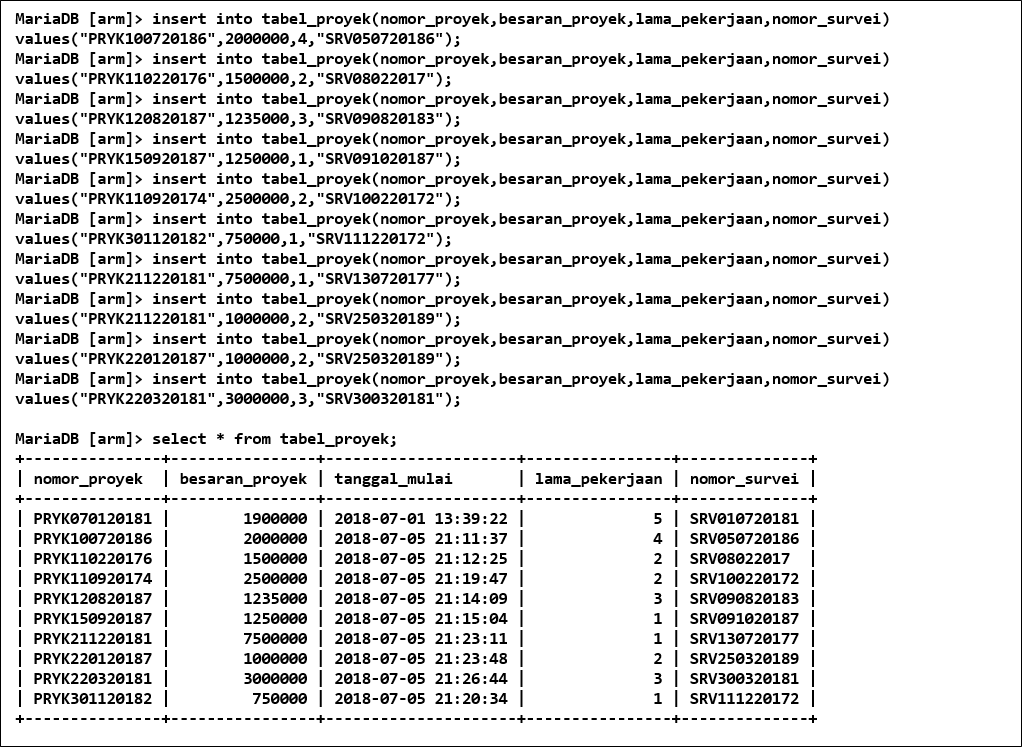
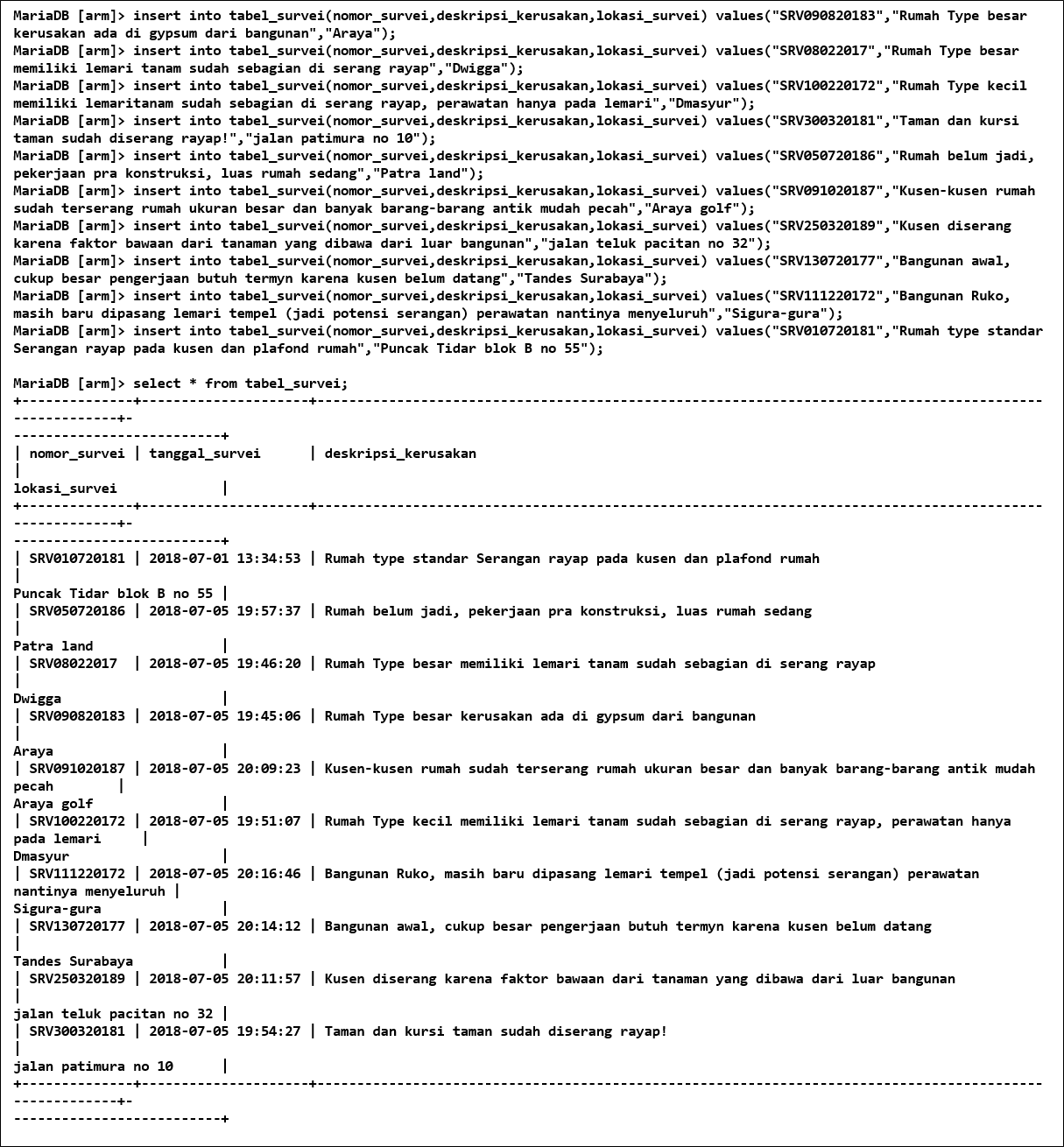
Gambar 4.9 memasukkan data ke tabel\_gudang beserta hasilnya

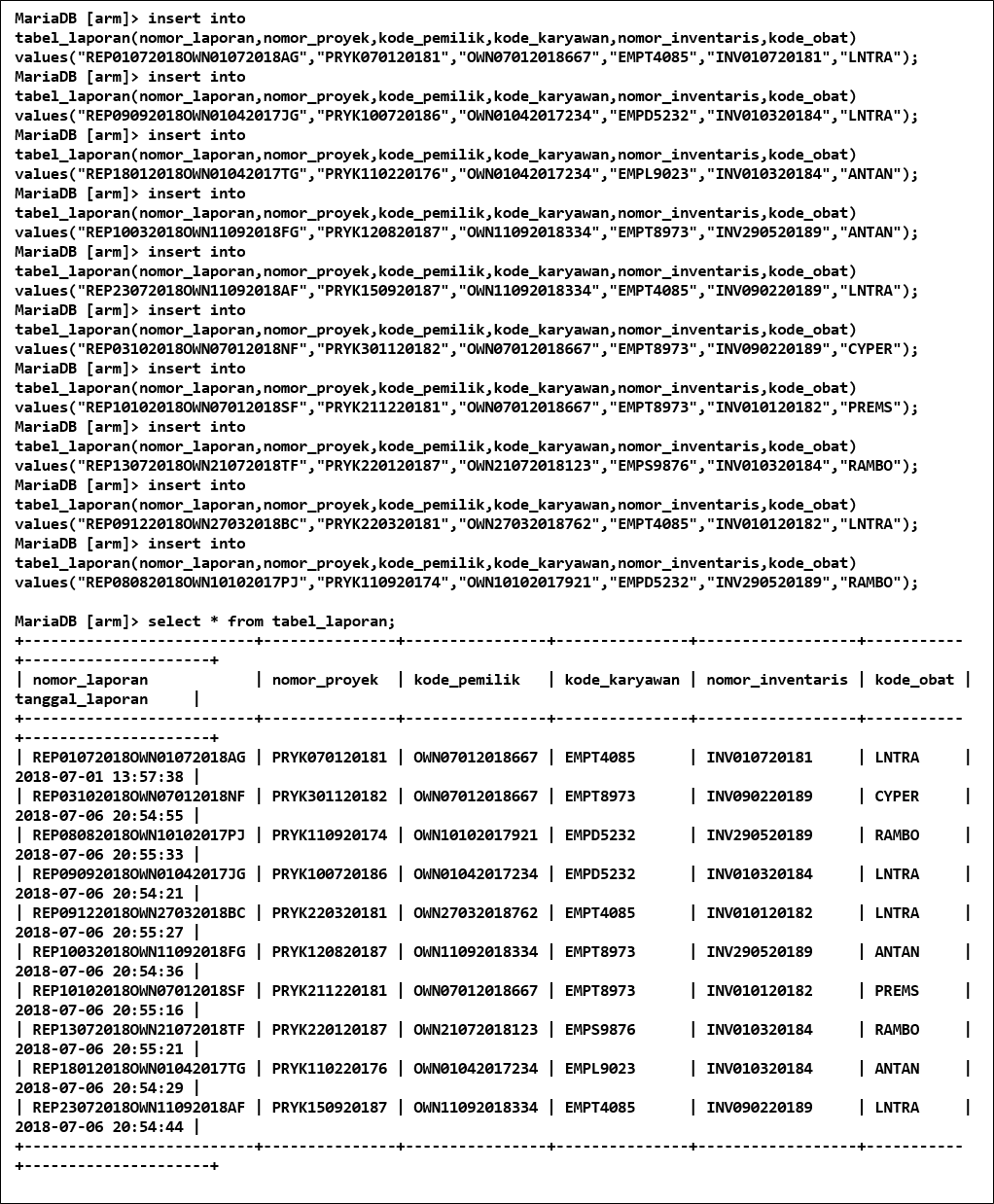
Gambar 4.10 memasukkan data ke tabel\_karyawan beserta hasilnya



Gambar 4.11 memasukkan data ke tabel\_inventaris beserta hasilnya

Gambar 4.12 memasukkan data ke tabel\_pemilik beserta hasilnya

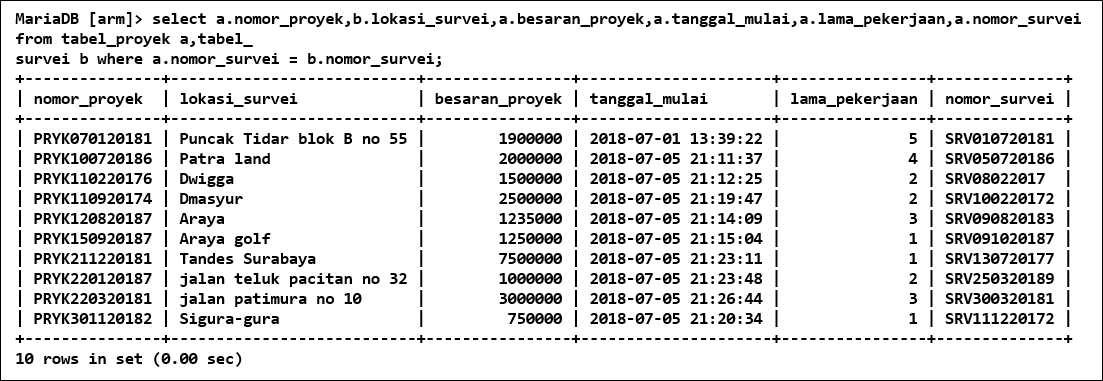
Gambar 4.13 memasukkan data ke tabel\_proyek beserta hasilGambar 4.14 memasukkan data ke tabel\_proyek beserta hasilnya

Gambar 4.15 memasukkan data ke tabel\_laporan beserta hasilnya

1. *Query* ambil data dari *database*

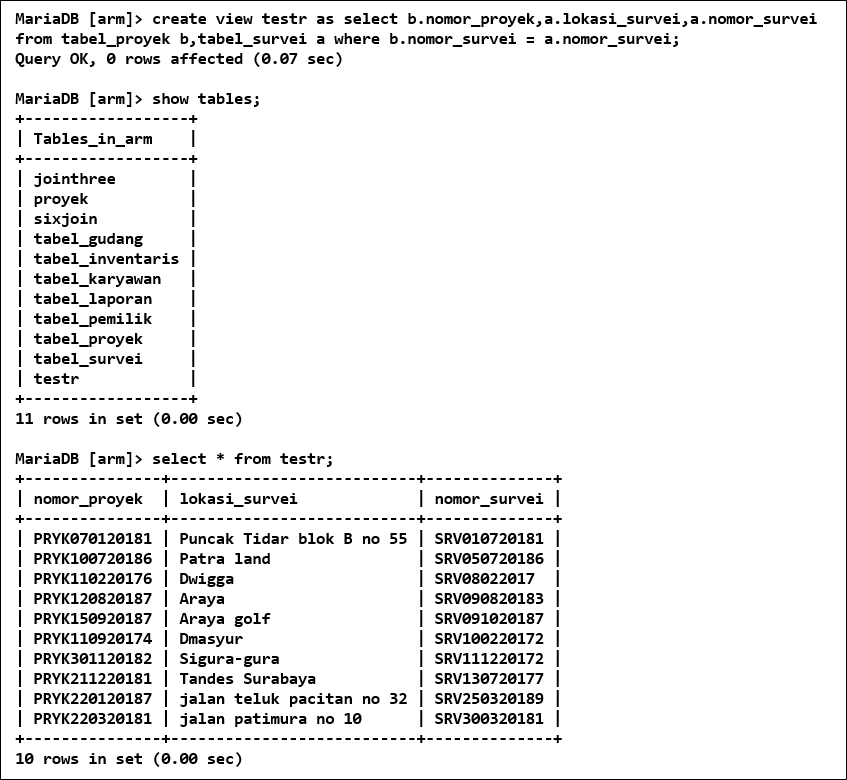
Sebelum kami membuat program sederhana untuk melihat data menggunakan bahasa pemprograman PHP, kami menguji beberapa query untuk mendapatkan data yang kami butuhkan. Beberapa *query* sudah kami sebutkan pada poin-poin pembahasan sebelum ini. Pada bagian ini kami hanya akan menjelaskan bagian *query* yang mneggunakan *join* dalam pengambilan datanya.

*Query* di bawah berfungsi untuk mengambil data pada tabel\_proyek dan tabel\_survei yang memiliki relasi sesuai dengan pilihan data dan parameter yang telah diberikan.

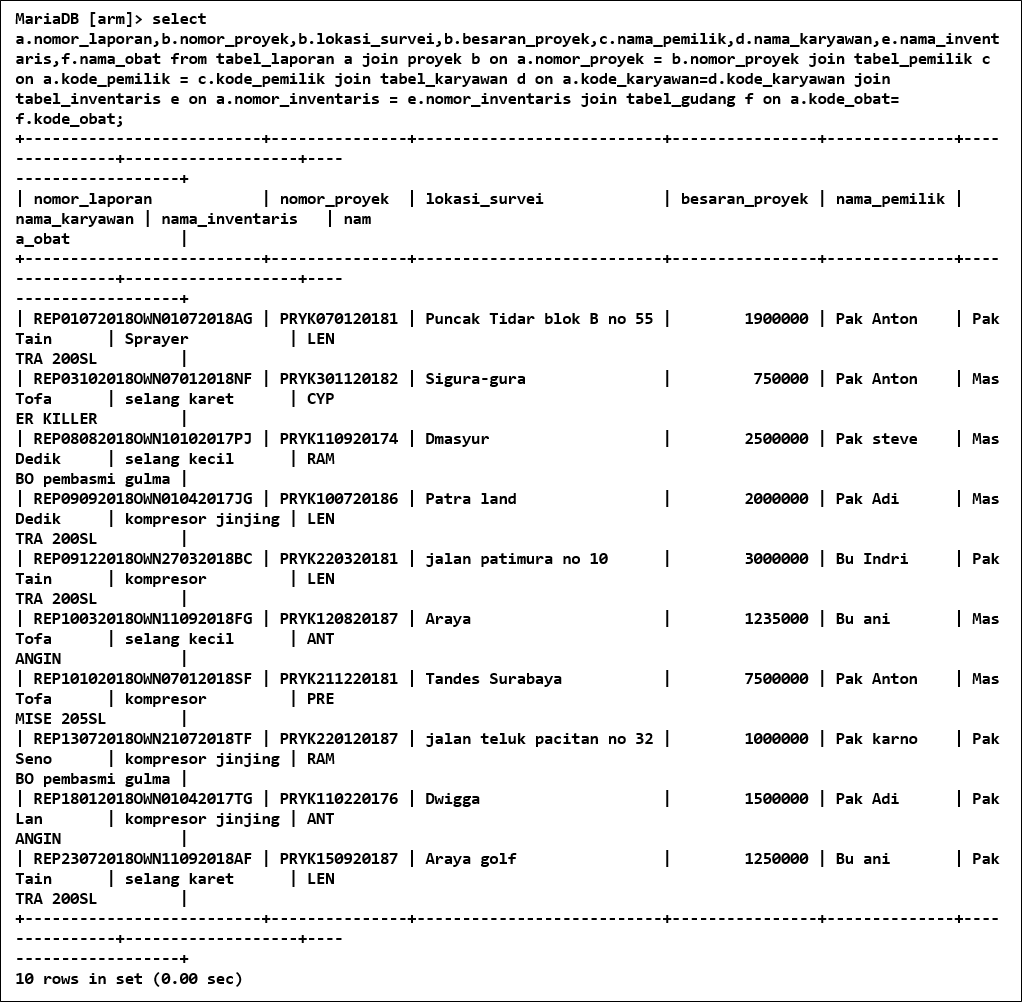


Gambar 4.16 mengambil data dari tabel dengan metode join

Selanjutnya kami menggunakan tabel\_laporan yang berfungsi untuk mengambil data dari tabel-tabel lain yang ada menggunakan parameter *foreign key* yang telah diberikan. Karena kami melakukan *join* antar tujuh tabel dan ada satu tabel yang tidak memiliki relasi langsung dengan tabel laporan, maka kami membuat sebuah *view* yang kemudian kami lakukan metode join antara hasil view dan tabel sisa yang belum direlasikan seperti pada gambar di halaman selanjutnya.

Gambar 4.17 membuat *view* dari sebuah *query join* 

Setelah berhasil membuat *view*, kami melanjutkan dengan melakukan proses join antara *view* yang kami buat dengan tabel-tabel yang lainya secara langsung. Hal ini kami lakukan untuk mengambil beberapa data yang terpisah di tabel-tabel yang berbeda pada *database.* Sehingga diperlukan untuk memanggil beberapa data yang letaknya ada di tabel yang berbeda-beda. Metode ini mempermudah pengambilan data dan mengurangi kejanggalan pada data sesuai dengan normalisasi yang dibuat. Semakin banyak data, proses *join* juga akan semakin berat, jadi perlu hati-hati untuk penggunaan *join* pada mysql



Gambar 4.18 *query join* untuk beberapa tabel sekaligus

Kemudian semua *query* yang kami tampilkan juga kami implementasikan pada program php yang telah kami buat yang nanti juga kami lampirkan pada laporan ini.

# **BAB V**

# **Kesimpulan dan Saran**

1. **Kesimpulan**

Setelah melakukan analisis mendalam terhadap usaha Anti Rayap Malang .com kami menemukan beberapa kendala yang sering dihadapi oleh usaha-usaha kecil yang sedang berkembang. Masalah utama yang mereka hadapi adalah keterbatasan sumber daya dan informasi untuk mengelola data yang mereka miliki secara utuh dan lebih efisien. Hal ini berdampak pada daya saing yang dimiliki oleh perusahaan tersebut, pengelolaan data yang baik akan mengarahkan usaha tersebut pada pengambilan keputusan yang lebih baik dan lebih menguntungkan pula untuk perusahaan tersebut. Namun sayangnya banyak usaha-usaha yang sedang berkembang tidak memperhatikan hal tersebut dan cenderung melambat pertumbuhannya.

Sehingga dari laporan yang kami buat, kami dapat mengambil kesimpulan bahwa penting untuk sebuah usaha memiliki sebuah sistem pengolahan data yang terpusat dan menyeluruh agar usaha tersebut dapat memanfaatkan data yang dimiliki dengan lebih maksimal dan menguntungkan. Hal ini juga dilandasi karena pengetahuan pelaku usaha menegah kebawah tentang dampak positif pemanfaatan teknologi untuk kemajuan sebuah usaha sangat minim dan cederung diabaikan.

1. **Saran dan Kritik**
2. Saran

Dari pemaparan yang kami sampaikan pada bab sebelumnya menunjukkan bahwa apa yang kami kerjakan berpotensi untuk dikembangkan dengan nilai ekonomis terttentu. Sehingga kami berharap model penugasan seperti yang kami terima dapat dilanjutkan dan terus menerus diperbaharui dasarnya. Agar ide-ide yang kami sampaikan dapa terwujud dan menjadi sesuatu yang memiliki nilai ekonomis juga.

1. Kritik

Dalam pengerjaan tugas ini kami mengalami kesulitan untuk mengelola pencapaian yang sudah kami dapat pada setiap babnya. Menyebabkan kami sedikit teledor dan melenceng dari rencana. Kedepannya akan lebih baik jika diharuskan setiap kelompok membuat suatu tabel *progress* yang dapat diamati prosentase pengerjaan dan kontribusi setiap kelompoknya.