

BAB I

PENDAHULUAN

1.1 Latar Belakang

PT. Lelco Trindo Graha Nusantara merupakan perusahaan yang bergerak di bidang pemasaran produk lampu, seperti lampu mekanik, lampu elektrik, dan sistem pengontrolannya.

Untuk memformalkan kebijakan keamanan dalam organisasi, PT. Lelco Trindo Graha Nusantara menggunakan kontrol akses Matrix Model, yang menentukan apa yang boleh dan yang tidak boleh diakses oleh pengguna. Untuk mengatur fungsi-fungsi dalam sistem informasi perusahaan yang berhubungan dengan keamanan sistem diserahkan kepada seorang administrator yang dipegang oleh divisi IT (*Information Technology*).

Kemungkinan ancaman yang dilihat penulis dari *user access* ini adalah keamanan data kata kunci (*password*) yang disimpan pada basis data di penyedia (*server*). Melihat permasalahan yang ada, penulis mencoba untuk membuat enkripsi terhadap data kata kunci yang disimpan agar tidak dapat dibaca dan dipergunakan oleh pihak yang tidak berhak.

Untuk melakukan ini ada suatu cara yang disebut penyandian data. Dengan kriptografi, data dapat diubah menjadi sandi-sandi yang tidak dimengerti serta mengembalikannya kembali ke semula, proses ini disebut enkripsi dan dekripsi data.

Dalam penyusunan tugas akhir ini penulis akan mengimplementasikan algoritma MD5 untuk enkripsi terhadap kata kunci yang ada pada basis data *user access*. Kunci yang digunakan dalam proses enkripsi dan dekripsi pada algoritma MD5 adalah sama.

1.2 Ruang Lingkup

Ruang lingkup yang akan dibahas pada laporan ini mencakup perancangan *user access* dan keamanan data kata kunci pada basis data *user access* dimana enkripsi kata kunci dilakukan saat proses penyimpanan pada basis data di penyedia. Enkripsi yang dilakukan penulis akan mengimplementasikan algoritma MD5.

1.3 Tujuan Penulisan

Tujuan penulisan dalam laporan tugas akhir ini adalah:

1. Mempelajari dan mengimplementasikan algoritma MD5 yang merupakan salah satu algoritma kriptografi modern untuk pengamanan data kata kunci pada basis data *user access*.
2. Melindungi data-data penting perusahaan dari orang-orang yang tidak memiliki hak akses.

1.4 Batasan Masalah

Ruang lingkup permasalahan yang akan dibahas mengenai sistem keamanan pada PT. Lelco Trindo Graha Nusantara, yaitu hanya terbatas pada masalah-masalah yang berhubungan dengan sistem keamanan *user access* yang mekanisme dan metode MD5 (*Message Digest Algoritma 5*) yang dipergunakan para

administrator/manager untuk mengontrol apa yang boleh diakses *user*, termasuk apa yang boleh dilakukan setelah otentikasi dan otorisasi, termasuk pemantauannya.

1.5 Metodologi

Dalam membuat aplikasi ini penulis menggunakan metode Waterfall, yaitu teknik dengan pendekatan matematis dan berurutan di dalam pengembangan sistem perangkat lunaknya.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini akan disusun menjadi lima bab dengan rinciannya sebagai berikut :

Bab I: Pendahuluan

Pada bab ini akan diuraikan latar belakang, ruang lingkup, tujuan penulisan, batasan masalah, metodologi, dan sistematika penulisan.

Bab II: Landasan Teori

Dalam bab ini penulis menguraikan teori-teori untuk mendukung permasalahan yang ada seperti waterfall, gambaran umum mengenai kriptografi, enkripsi, algoritma MD5, dan teori dasar *Unified Modeling Language* (UML) serta PHP sebagai bahasa pemrogramannya dan MySQL.

Bab III: Analisis dan Perancangan Program *User Access* pada PT. Lelco Trindo Graha Nusantara.

Bab ini berisi langkah-langkah dalam pembuatan sistem keamanan *user access* pada PT. Lelco Trindo Graha Nusantara dengan bahasa

pemrograman PHP (*Pre Hypertext Preprocessor*) dan metodologi enkripsi yang digunakan adalah MD5 (*Message Digest Algorithm 5*).

Bab IV: Implementasi dan Pengujian

Dalam bab ini penulis mencoba mengimplementasikan dan menguji fungsi-fungsi program yang penulis buat.

Bab V : Penutup

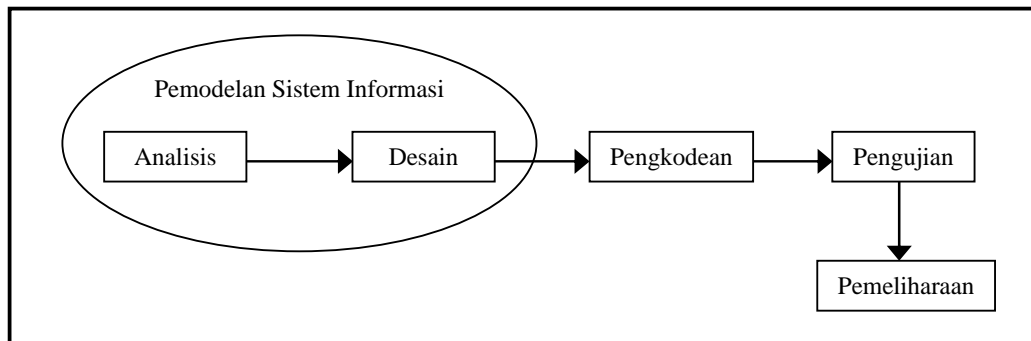
Dalam bab ini penulis mencoba mengambil kesimpulan berdasarkan analisis-*analisis* yang dilakukan. Selain itu penulis juga memberikan saran yang kiranya dapat bermanfaat bagi para *programmer* yang sedang mempelajari pembuatan program keamanan berbasis *web* dengan menggunakan bahasa pemrograman PHP (*Pre Hypertext Preprocessor*).

BAB II

LANDASAN TEORI

2.1 Rekayasa Perangkat Lunak Model Sekuensial Linier

Rekayasa perangkat lunak model ini sering juga disebut dengan siklus kehidupan klasik atau model air terjun (*Waterfall*)



Gambar 2.1 Model Sekuensial Linier (Pressman, 2002: 37)

Sekuensial Linier menerapkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial. Dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Model Sekuensial Linier melingkupi aktivitas- aktivitas sebagai berikut :

2.1.1 Rekayasa dan Pemodelan Sistem / Informasi

Karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut.

Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak itu sendiri, *user*, dan data yang diolah. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan jumlah kecil analisis serta desain tingkat puncak.

2.1.2 Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan diintensifkan dan di fokuskan khususnya pada perangkat lunak. Untuk membangun sifat program yang dibangun, perekayasa perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antar muka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat kembali bersama dengan *user*.

2.1.3 Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, yaitu struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat/kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

2.1.4 Pengkodean

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan

2.1.5 Pengujian

Setelah kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

2.1.6 Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan pada user. Perubahan akan terjadi karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya tanpa perlu membuat yang baru lagi.

2.2 Teori Dasar Pengamanan Data

Keamanan merupakan faktor yang sangat penting dan tidak dapat diabaikan dalam menyembunyikan *source code* suatu program atau aplikasi dan mengirimkan data pada jaringan internet. Serangan-serangan yang dilakukan untuk membongkar, mengacak, melihat bahkan mencuri isi dari suatu *source code* mempunyai banyak bentuk misalnya mendekompilasi *file*, memodifikasi *source code*, dan lainnya. Jika

di dalam suatu sistem keamanan terdapat vurnerabilitas yang dapat diekplotasi, maka akan menyebabkan sistem keamanan tersebut muah diserang oleh berbagai ancaman. Ancaman terhadap jaringan internet dapat disebabkan oleh manusia, bencana alam, kesalahan manusia dan kerusakan perangkat keras atau perangkat lunak.

Kemajuan dibidang telekomunikasi dan komputer telah memungkinkan seseorang untuk melakukan transaksi bisnis secara *realtime*, selain itu ia juga dapat mengirimkan informasi kepada temannya secara *online*. Kegiatan-kegiatan tersebut tentu saja akan menimbulkan resiko bilamana informasi yang sensitif dan berharga tersebut dapat diakses oleh orang-orang yang tidak berhak (*anauthorized persons*).

Dalam era konektifitas elektronik universal, *hacker*, virus, penipuan elektronik maupun diam-diam secara elektronik, maka keamanan data benar-benar menjadi permasalahan yang sangat penting. Perkembangan sistem komputer dan interkoneksinya melalui jaringan telah meningkat, tentu saja hal ini membutuhkan keamanan data yang handal agar terhindar dari serangan. Untuk mengamankan data di jaringan diperlukan kriptografi dengan metode enkripsi.

Dalam menjaga kerahasiaan data, kriptografi mentransformasikan data jelas (*plaintext*) kedalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. *Ciphertext* inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, *ciphertext* tersebut ditrasformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali.

S. Ganfinkel (2002) berpendapat bahwa karakteristik keamanan komputer dikatakan aman berkaitan dengan tiga tujuan berikut ini yaitu:

1. Kerahasiaan, berarti sebuah aset dari sistem komputer yang dapat diakses hanya oleh pihak yang berhak atau legal.
2. Integritas, berarti sebuah aset hanya bisa dimodifikasi oleh pihak-pihak yang berhak atau dengan cara yang legal.
3. Ketersediaan, berarti sebuah aset dapat diakses oleh orang yang berhak saja dimana tidak dibatasi hanya mengakses suatu bagian saja.

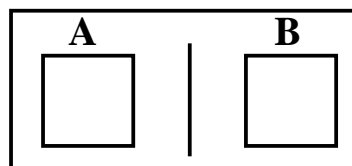
2.2.1 Ancaman Terhadap Keamanan Data

Pengamanan data dalam komunikasi menjadi hal yang sangat penting dalam pengembangannya, karena dengan semakin meningkatnya penggunaan komputer diberbagai bidang dan sektor. Dengan semakin banyaknya data yang diproses, yang kemudian dikirim melalui media perangkat komunikasi elektronik, maka semakin besar ancaman yang dapat terjadi terhadap data tersebut.

Ada empat bentuk ancaman yang dapat terjadi terhadap data komputer, yaitu:

1. Interupsi (*Interruption*)

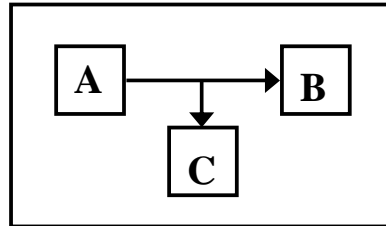
Interupsi adalah jika ada suatu data yang dikirimkan tidak sampai kepada pihak yang berhak atas data yang telah dikirimkan. Interupsi juga merupakan suatu bentuk penyerangan terhadap sifat ketersediaan data (*availability*). Contohnya adalah kerusakan pada perangkat keras (*hardware*), kegagalan sistem operasi (*operating system*), sehingga tidak dapat menemukan file yang dicari.



Gambar 2.2 Interupsi

2. Intersepsi (*Interception*)

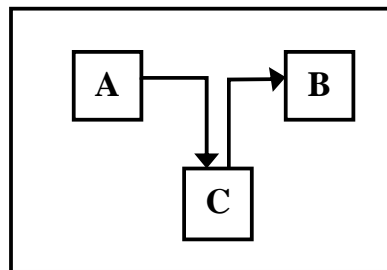
Intersepsi adalah jika pihak ketiga berhasil mendapatkan data yang dikirimkan. Intersepsi ini juga merupakan suatu bentuk penyerangan terhadap sifat kerahasiaan data (*confidentiality*). Contohnya adalah penggandaan program atau *file* data yang tidak terlihat, atau pencurian data pada jaringan dengan cara *widetrapping*.



Gambar 2.3 Intersepsi

3. Modifikasi (*Modification*)

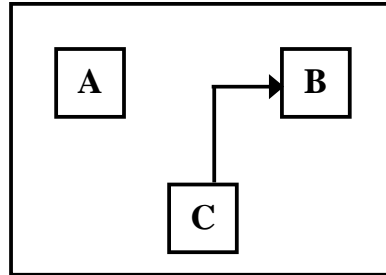
Modifikasi adalah berhasilnya pihak ketiga mengubah data yang dikirimkan. Modifikasi juga merupakan suatu bentuk penyerangan terhadap sifat keaslian data (*authenticity*). Contohnya adalah merubah program.



Gambar 2.4 Modifikasi

4. Fabrikasi (*Fabrication*)

Fabrikasi adalah berhasilnya penyerang mengirimkan data dengan memanfaatkan identitas orang lain. Fabrikasi juga merupakan bentuk penyerangan terhadap sifat keaslian data (*authenticity*).



Gambar 2.5 Fabrikasi

Dengan semakin meningkatnya komunikasi data yang bersifat rahasia, seperti pengiriman dokumen-dokumen rahasia, maka data perlu dilindungi agar kerahasiaannya tetap terjaga. Jika data tidak dilindungi keamanannya, maka pihak lain akan dapat membacanya dan menjadi ancaman yang merugikan. Untuk mengantisipasi bentuk-bentuk ancaman tersebut, maka perlu dilakukan suatu cara pencegahan untuk melindungi data yang akan dikirimkan. Salah satu cara yang dilakukan adalah dengan melakukan pengkodean pada data yang dikirimkan. Dapat juga dilakukan dengan menyembunyikan data kedalam bentuk tertentu, sehingga data tidak dapat dimengerti oleh pihak lain yang tidak diinginkan. Cara-cara yang dilakukan ini merupakan bagian dari kriptografi.

2.3 Pengertian Kriptografi

Kriptografi adalah metode penyimpanan data, baik dalam jangka waktu yang pendek maupun panjang, ke dalam bentuk yang dapat diketahui pihak yang berhak, tetapi disembunyikan dari pihak lain. Kriptografi juga merupakan ilmu pemetaan dari teks yang tidak dapat dimengerti oleh manusia, ke dalam bentuk lain yang tidak dapat dimengerti oleh manusia. Tujuan dari pemetaan tersebut adalah untuk mempengaruhi tampilan atau penampakan dari teks, tapi tidak merubah arti dari teks

tersebut. Kriptografi dapat diartikan ilmu yang mempelajari penyajian data, saat pengiriman data dan pemulihan data ketika diterima oleh pihak yang berhak.

Kriptografi terdiri dari 2 (dua) proses dasar yang saling terkait, yaitu proses enkripsi dan proses dekripsi. Kriptografi memiliki tujuan utama, yaitu untuk melindungi data awal (*plaintext*) dari pihak yang tidak berkepentingan.

Kata kriptografi, secara etimologi (ilmu asal usul kata), berasal dari dua kata dalam bahasa Yunani, yaitu *kryptos* dan *graphia*. Kata *kryptos* bermakna sesuatu yang disembunyikan, tidak dikenal, diselubungi, rahasia atau misterius, dan kata *graphia* bermakna tulisan. Jadi secara keseluruhan, kriptografi dapat diartikan sebagai suatu seni atau ilmu dari tulisan rahasia. Kriptografi berarti teknik perubahan pesan asli menjadi pesan yang sudah dimodifikasi atau dapat diartikan metode untuk menyembunyikan suatu pesan rahasia dengan menggunakan persamaan matematis untuk melakukan enkripsi. Enkripsi digunakan untuk menyembunyikan informasi asli oleh pemakai yang berhak ke dalam bentuk yang tidak dapat dimengerti.

Perkembangan internet yang semakin pesat, membuat aspek keamanan harus selalu mengikutinya. Semakin canggih sistem keamanan internet, semakin canggih pula orang-orang yang berusaha untuk memecahkan sistem keamanan tersebut. Oleh karena itu semakin hari sistem keamanan terus dikembangkan dan ditingkatkan.

Transaksi bisnis yang semakin banyak dilakukan melalui Internet memancing setiap orang untuk berusaha mencuri informasi yang ada untuk kepentingan pribadi. Ada yang menyerang melalui perangkat keras maupun perangkat lunak. Orang yang tidak berhak terhadap informasi tersebut atau yang sering disebut *hacker* semakin sulit untuk dilacak karena internet merupakan jaringan maya.

Salah satu metode keamanan yang sudah banyak dilakukan adalah dengan kriptografi. Kriptografi adalah ilmu yang mempelajari bagaimana membuat suatu pesan yang dikirim pengirim dapat disampaikan kepada penerima dengan aman.

Ada tiga kriteria umum yang dipergunakan untuk memecahkan dan menentukan keamanan dari sebuah algoritma kriptografi, yaitu:

1. Nilai usaha, algoritma yang relatif aman bila nilai usaha yang dibutuhkan lebih besar dari nilai informasi yang terenkripsi.
2. Algoritma yang relatif aman, bila waktu yang dibutuhkan lebih lama dari waktu kadaluarsa informasi yang terenkripsi.
3. Jumlah data, algoritma yang relatif aman bila jumlah data yang di butuhkan lebih besar dari data yang terenkripsi.

2.3.1 Macam-macam Kriptografi

Kriptografi dibedakan menjadi tiga bagian yaitu sebagai berikut:

1. Kriptografi simetris

Fungsi kriptografi simetris disebut kriptografi kunci rahasia merupakan jenis kriptografi yang paling intuitif. Ini termasuk penggunaan kunci rahasia yang dikenal hanya pada pengguna komunikasi yang aman.

2. Kriptografi Asimetris

Berbeda dengan kriptografi simetris, kriptografi asimetris menggunakan dua kunci yang berbeda, yaitu kunci publik dan kunci pribadi. Kunci-kunci itu berhubungan secara matematis, tetapi tidak mungkin secara perhitungan untuk menarik kesimpulan satu dengan yang lain.

3. Fungsi *hash* satu arah

Fungsi *hash* atau satu arah, juga dikenal sebagai rangkuman pesan atau fungsi kompresi adalah fungsi matematis yang mengambil input panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap.

2.4 Enkripsi dan Dekripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *chipper*. Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak. Informasi ini dapat berupa nomor kartu kredit, catatan penting dalam komputer, maupun password untuk mengakses sesuatu. Sedangkan Dekripsi adalah kebalikan dari proses enkripsi yaitu proses konversi data yang sudah di enkripsi (*chiphertext*) kembali menjadi data aslinya (*original plaintext*) sehingga dapat dibaca atau dimengerti kembali.

Salah satu strategi yang paling sering digunakan dalam keamanan jaringan adalah enkripsi data dan erat kaitannya dengan tanda tangan digital (*digital signature*). *Digital signature* menggunakan prinsip seperti tanda tangan manusia pada lembar dokumen dan dilakukan secara digital. Ia menyatakan keabsahan si pengirim data bahwa data yang dikirimkan benar-benar berasal dari si pengirim. Pada saat ini, enkripsi data dapat dilakukan di perangkat lunak maupun di perangkat keras. Berbagai jenis metode enkripsi data pada tingkat aplikasi telah dikembangkan seperti RSA, MD-5, IDEA, SAFER, Skipjack, Blowfish, dan lain sebagainya. Dengan strategi ini, transfer data ke jaringan komputer berlangsung secara konfidensial.

Model-model enkripsi beserta algoritma yang akan dipakai untuk setiap enkripsi ada 2 hal yang penting dan akan dijabarkan, yaitu sebagai berikut:

1. Enkripsi dengan kunci pribadi atau rahasia

Enkripsi dapat dilakukan jika si pengirim dan si penerima telah sepakat untuk menggunakan metode enkripsi atau kunci enkripsi tertentu. Metode enkripsi atau kuncinya ini harus dijaga supaya tidak ada pihak luar yang mengetahuinya. Cara enkripsi ini juga dikategorikan sebagai kriptografi simetris.

2. Enkripsi dengan kunci publik

Cara enkripsi ini mempunyai banyak kelebihan, salah satunya adalah tiap orang hanya perlu memiliki satu set kunci, tanpa peduli berapa banyak orang yang akan di ajak berkomunikasi. Cara enkripsi ini tidak membutuhkan saluran yang aman untuk pengiriman kunci, sebab kunci yang dikirim ini memang harus diketahui oleh publik. Cara enkripsi ini sangat praktis sehingga masyarakat umum pun dapat dengan mudah memakainya.

Ada dua cara untuk menghasilkan kunci enkripsi, yaitu sebagai berikut:

1. Kunci dikenal fungsi *hash* dengan menggunakan MD-5 dan disimpan dalam sektor yang sudah disiapkan dalam hard disk.
2. Kunci tidak boleh disimpan disembarang tempat. Dalam hal ini pengguna harus memasukkan kunci pada waktu *startup*. Ini adalah cara yang paling aman, tetapi pengguna tidak bisa melupakan kunci karena terjadi hal ini maka akan kesulitan waktu *startup*.

2.5 Teori Metode MD5

Algoritma MD5 mengambil sebuah input sebuah pesan yang panjangnya berbeda-beda dan menjadikannya output 128-bit berupa “sidik jari” atau “intisari pesan” dari input tersebut. Algoritma MD5 dimaksudkan bagi aplikasi pengenalan digital, yang mana file-file besar harus di “*compressed*” dengan cara yang benar sebelum di enkripsi dengan private (rahasia) key dibawah *cryptosystem public key* seperti RSA.

Algoritma MD5 adalah ekstensi dari algoritma *message-digest* MD4 MD5 sedikit lebih lambat dari MD4 tetapi mempunyai sifat desain yang lebih konservatif daripada MD4. MD5 dirancang karena MD4 dirasa lebih peruntukkan untuk keperluan kinerja yang lebih cepat tanpa memperhatikan *critical review*, karena MD4 dirancang untuk menjadi sangat cepat sehingga beresiko membahayakan serangan *cryptanalytic* yang sukses, sedangkan MD5 dengan kecepatan yang agak lambat memberikan keamanan yang lebih kuat. MD5 ditempatkan pada *domain public* untuk peninjauan dan kemungkinan adopsi metode sebagai standar.

2.5.1 Algoritma MD5

Dokumen ini menjelaskan tentang algoritma MD5 (*message-digest algorithm*). Algoritma MD5 mengambil input dengan panjang tertentu dan mengubahnya menjadi keluaran berupa “sidik jari” atau *Message digest* sebesar 128 bit. Keluaran dari input tersebut lalu dibuat perkiraan yang tak terlihat secara perhitungan untuk membuat dua pesan yang mempunyai intisari yang sama. Algoritma MD5 dimaksudkan untuk membuat tanda tangan aplikasi, dimana sebuah file yang besar dapat di “compressed” dengan cara yang aman sebelum di enkripsi

dengan kunci *private* (rahasia) dibawah sebuah *public-key cryptosystem* seperti RSA.

Algoritma MD5 dirancang untuk dapat berjalan cepat pada mesin 32-bit. Sebagai tambahan, Algoritma MD5 tidak memerlukan table pengganti beukuran besar; algoritma dapat dikodekan secara cukup padat.

Algoritma MD5 adalah kelanjutan dari algoritma *message-digest* MD4. MD5 sedikit lebih lambat dari MD4, tetapi mempunyai desain lebih konservatif.

2.5.2 Deskripsi algoritma MD5

Kita mulai dengan memasukkan input dengan ukuran b -bit da kita ingin mencari message digestnya. Ini adalah perkiraan dari integer non negatif; b dapat berupa nol, b tidak haus berupa angka kelipatan angka delapan, dan dapat mempunyai ukuran yang besar. Kita bayangkan bit dari pesan seperti berikut:

$m_0 m_1 \dots m_{\{b-1\}}$

Berikut adalah lima langkah yang dilakukan untuk menghitung *message digest*:

1. Langkah pertama menambahkan bits pengisi (*Padding*)

Message tersebut "ditambah" (diperluas) sehingga memiliki panjang (dalam bits) sehingga sesuai dengan 448, modul 512. Apabila, *message* tersebut diperluas sehingga hanya menjadi 64 bit daripada digandakan menjadi panjang 512 bit. Penambahan selalu dilaksanakan, meskipun panjang dari message tersebut sudah sesuai dengan 448, modul 512.

Padding dilakukan sebagai berikut: sebuah "1" bit di *appended* (ditambahkan) kedalam message, lalu "0" bits ditambahkan sehingga panjang dalam bits message yang ditambahkan menjadi sesuai dengan 448, modul 512. secara keseluruhan, setidak-tidaknya satu bit dan sebanyak-banyaknya 512 bit ditambahkan.

2. Langkah kedua Panjang Append (penambahan).

Sebuah 64-bit merupakan representasi dari b (panjang message sebelum padding (penambahan) dilakukan) ditambahkan ke hasil dari langkah sebelumnya. Dalam peristiwa yang tidak lazim jika besar b lebih besar dari 2^{64} , lalu hanya urutan terendah dari b yang digunakan. (bit di ditambah sebagai dua kata sebesar 32-bit dan ditambahkan kata urutan-rendah terlebih dahulu menurut ketentuan sebelumnya.)

Pada poin ini hasil message (setelah penambahan dengan bits dan b) mempunyai panjang kelipatan tepat 512 bits. Secara ekuivalen, message ini memiliki panjang kelipatan kata tepat dari 16 (32-bit). dimana $M[0 \dots N-1]$ menunjukkan kata dari hasil message, dimana N adalah kelipatan dari 16.

3. Langkah ketiga. Menginisialisasi (Mengenali) Buffer Message Digest

Sebuah buffer empat huruf (A,B,C,D) digunakan untuk menghitung message digest. Dimana tiap kata dari A, B, C, D adalah register 32-bit. Register ini diinisialisasi ke nilai hexadecimal berikut.

Dimulai dari low-order bytes):

kata A: 01 23 45 67

kata B: 89 ab cd ef

kata C: fe dc ba 98

kata D: 76 54 32 10

4. langkah keempat Process Message dalam blok 16-kata

Pertama kita mendefinisikan empat fungsi auxiliary yang dimana tiap fungsi adalah input.

Tiga kata 32-bit dan menghasilkan keluaran satu kata 32-bit.

$$F(X,Y,Z) = XY \vee \text{not}(X) Z$$

$$G(X,Y,Z) = XZ \vee Y \text{not}(Z)$$

$$H(X,Y,Z) = X \text{ xor } Y \text{ xor } Z$$

$$I(X,Y,Z) = Y \text{ xor } (X \vee \text{not}(Z))$$

Tiap posisi bit F berfungsi sebagai conditional: if X then Y else Z.

Fungsi F dapat didefinisikan menggunakan + bukan \vee since XY

and not(X)Z tidak akan menempati posisi pertama dalam posisi bit yang sama.) harus dicatat, jika bit X, Y, dan Z berdiri sendiri dan tidak memihak setiap bit F(X,Y,Z) akan berdiri sendiri dan tidak memihak.

Fungsi G, H, dan I sama dengan fungsi F, dalam hal ini berperan dalam "bitwise parallel" untuk menghasilkan output dari bit X, Y, dan Z, seperti bit X, Y, dan Z berdiri sendiri dan tidak memihak, dan tiap bit G(X,Y,Z), H(X,Y,Z), dan I(X,Y,Z) akan berdiri sendiri dan tidak memihak.

Catatan: Fungsi H bit-wise "xor" atau "parity" function dari inputs. Langkah ini menggunakan tabel 64-element T[1 ... 64] dibangun dari fungsi sinus.

Let T[i] denote elemen ke-i dari tabel, yang sama dengan integer part of 4294967296 times abs(sin(i)), dimana i dalam bentuk radians. Elemen dari tabel di berikan secara appendix.

Contoh:

```
/* Process each 16-word block. */
```

```
For i = 0 to N/16-1 do
```

```
/* Copy block i into X. */
```

```
For j = 0 to 15 do
```

```
Set X[j] to M[i*16+j].
```

```

end /* of loop on j */

/* Save A as AA, B as BB, C as CC, and D as DD. */

AA = A

BB = B

CC = C

DD = D

/* Round 1. */

/* Let [abcd k s i] denote the operation
a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */

/* Do the following 16 operations. */

[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

/* Round 2. */

/* Let [abcd k s i] denote the operation
a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */

/* Do the following 16 operations. */

[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

/* Round 3. */

/* Let [abcd k s t] denote the operation

```

```

a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */

/* Do the following 16 operations. */

[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]

/* Round 4. */

/* Let [abcd k s t] denote the operation

a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */

/* Do the following 16 operations. */

[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

/* Then perform the following additions. (That is increment each
of the four registers by the value it had before this block
was started.) */

A = A + AA
B = B + BB
C = C + CC
D = D + DD

end /* of loop on i */

```

5. langkah kelima, Output

message digest yang dihasilkan sebagai output adalah A, B, C, D. Itu jika kita mulai dari low-order byte A, dan mengakhiri dengan high-order byte D.

2.6 Unified Modeling Language (UML)

Unified Modeling Language selanjutnya disebut UML adalah suatu bahasa permodelan standar untuk menulis rancangan perangkat lunak. UML dapat digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi suatu perangkat lunak yang intensif dari suatu sistem. Sebuah pembuat perangkat lunak yang sehat atau normal paling tidak dapat menghasilkan dokumen persyaratan (*requirements*), arsitektur (*architecture*), pemodelan (*design*), kode program (*source code*), perencanaan proyek (*project plans*), pengujian (*tests*), prototipe (*prototypes*), dan penggunaan (*releases*).

UML memiliki tiga unsur penting, yaitu:

- a. Blok-blok bangunan, terdiri dari tiga jenis, yaitu berbagai hal, hubungan, dan diagram,
- b. Aturan yang mengatur bagaimana blok-blok itu dihubungkan,
- c. Mekanisme yang dapat digunakan.

Untuk memahami UML, perlu diketahui tiga karakteristik penting dari UML, yaitu:

1. Pengendali *use case* (*use case driven*), berarti *use case* digunakan sebagai awalan untuk membuat perilaku, verifikasi dan validasi arsitektur sistem. Selanjutnya *use case* digunakan untuk pengetesan sistem dan sebagai alat komunikasi di antara pihak-pihak yang berkepentingan dengan pembangunan sistem,

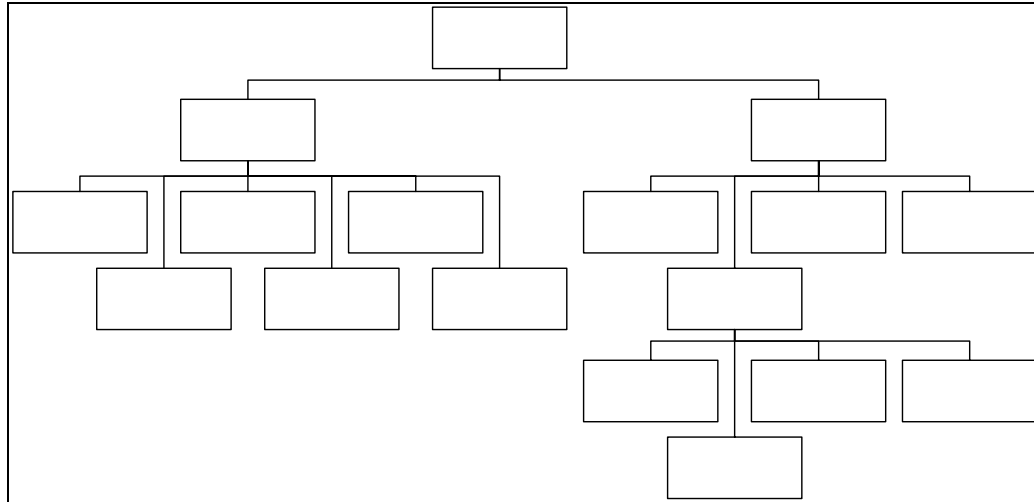
2. Arsitektur yang sentris (*architecture centric*), berarti arsitektur sistem digunakan sebagai pegangan utama untuk membuat konsep, mengkonstruksi, mengatur, dan menyusun sistem yang sedang dikembangkan,
3. Iterasi (*iterative*) dan proses penambahan (*incremental process*). *Iterative* berarti proses itu menyangkut pernyataan atau keputusan yang dapat dikerjakan secara berkelanjutan. *Incremental process* adalah satu proses yang melibatkan integrasi terus menerus dari arsitektur sistem untuk menghasilkan pernyataan atau keputusan tersebut diikuti oleh pernyataan atau keputusan berikutnya yang lebih baik dari sebelumnya. *Iterative* dan *incremental process* adalah pengendali resiko (*risk driven*), artinya pernyataan atau keputusan yang baru difokuskan untuk mengatasi atau mengurangi risiko yang paling besar untuk suksesnya sistem yang dibangun.

Penting sekali untuk membedakan antara UML dan diagram-diagram bagian dari sistem. Sebuah diagram adalah sebuah bagian yang mewakili dari suatu sistem model. Model juga mengandung sebuah backplane semantik yang tersimpan, seperti yang sudah tertulis dalam *use case*, yang mengarahkan pada model elemen dan diagram. Ada tiga bagian terkemuka (*prominent*) dari sistem model, yaitu:

1. Model fungsional (*functional model*), menunjukkan fungsi dari masalah suatu sistem dari sudut pandang pengguna, termasuk *use case* diagram.
2. Model objek (*object model*), menunjukkan masalah struktur dan sub-struktur dari sistem yang mempergunakan objek, atribut, operasi dan hubungan, termasuk diagram kelas (*class diagram*).

3. Model dinamis (*dynamic model*), menunjukkan masalah perilaku internal dari sistem, termasuk diagram urutan (*sequence diagram*), diagram aktifitas (*activity diagram*), dan diagram status mesin (*state machine diagram*).

Dalam UML 2.0 ada tiga belas jenis diagram. Untuk memahaminya, diagram tersebut dapat dikategorikan secara hirarki, seperti yang terlihat pada Gambar 2.6.



Gambar 2.6 Hirarki diagram UML (wikipedia.org)

Dari Gambar 2.6 dapat dijelaskan sebagai berikut:

1. Diagram struktur (*structure diagram*) menjelaskan apa yang harus ada pada

sistem yang dimodelkan, terdiri atas enam diagram, yaitu: **Structure Diagram**

a. Diagram kelas (*class diagram*),

b. Diagram komponen (*component diagram*),

c. Diagram struktur gabungan (*composite structure diagram*), **Class Diagram** **Component Diagram** **Object Diagram**

d. Diagram pengembangan (*deployment diagram*),

e. Diagram objek (*object diagram*), **Composite Structure Diagram**

f. Diagram paket (*package diagram*).

Diagram

Deployment Diagram

2. Diagram perilaku (*behavior diagram*) menjelaskan apa yang harus terjadi pada sistem yang dimodelkan, terdiri atas tiga diagram, yaitu:
 - a. Diagram aktifitas (*activity diagram*),
 - b. Diagram status mesin (*state machine diagram*),
 - c. Diagram *use case* (*use case diagram*).
3. Diagram interaksi (*interaction diagram*), merupakan bagian dari diagram tingkah laku, menunjukkan aliran kontrol dan data diantara proses-proses yang ada pada sistem yang dimodelkan, terdiri atas empat diagram, yaitu:
 - a. Diagram komunikasi (*communication diagram*),
 - b. Diagram ikhtisar interaksi (*interaction overview diagram*) pada UML 2.0,
 - c. Diagram urutan (*sequence diagram*),
 - d. Diagram pewaktuan UML (*UML timing diagram*) pada UML 2.0.

2.6.1 Diagram *use case*

Diagram *use case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem (artikel.webgaul.com). Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya.

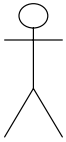


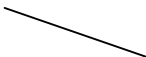
Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Diagram *use case* dapat sangat membantu bila kita sedang menyusun persyaratan sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang kasus pengujian untuk semua corak (*feature*) yang ada pada sistem.

Sebuah *use case* dapat meliputi (*include*) fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang ditampung akan dipanggil setiap kali *use case* yang meliputi dieksekusi secara normal. Sebuah *use case* dapat ditampung oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang umum (*common*).

Sebuah *use case* juga dapat meluas (*extend*) *use case* lain dengan tingkah lakunya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Tabel 2.1 menjelaskan empat komponen diagram *use case* yaitu:

Tabel 2.1 Notasi diagram *use case* (Djajono, 2005: 51-57)

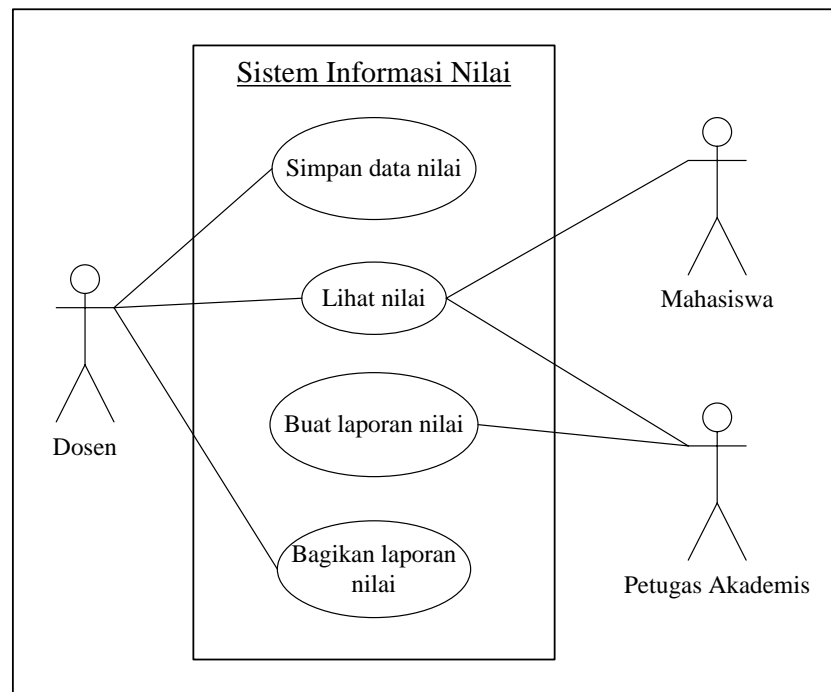
Notasi	Nama	Keterangan
	Aktor	Aktor adalah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Aktor diberi nama sesuai dengan tugasnya ketika menggunakan sistem. Jangan memberi nama aktor dengan nama orang. Aktor dapat berupa sistem lain di luar sistem yang dikembangkan
	Sistem	Sistem berupa ruang lingkup aplikasi yang dikembangkan.
	<i>Use case</i>	<i>Use case</i> adalah kegiatan atau fungsi yang dilakukan oleh aktor, diberi nama sesuai fungsi yang dilakukan oleh aktor.
	Hubungan (<i>relationship</i>)	Hubungan yang dilakukan oleh aktor dengan <i>use case</i> yang dilakukan oleh aktor tersebut.

Langkah-langkah membuat diagram *use case*:

1. Identifikasi semua aktor,
2. Identifikasi semua *use case*,
3. Urutkan prioritas *use case*,

4. Rincilah setiap *use case*,
5. Identifikasi adanya generalisasi tiap *use case*,
6. Identifikasi hubungan *include*,
7. Identifikasi hubungan *extend*,
8. Gambarkan diagram *use case*.

Contoh dari diagram *use case* digambarkan pada Gambar 2.7 berikut:



Gambar 2.7 Contoh diagram *use case* (Djajono, 2005: 54)

Dari Gambar 2.7, terdapat aplikasi Sistem Informasi Nilai yang dipergunakan oleh tiga orang aktor, yaitu Dosen, Mahasiswa, dan Petugas Akademis. Dosen dapat menyimpan data nilai, melihat nilai dan membagikan laporan nilai. Mahasiswa dapat melihat nilai. Petugas Akademis dapat melihat nilai dan membuat laporan nilai.

2.6.2 Diagram Aktifitas

Diagram aktifitas (*activity diagram*) menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir (artikel.webgaul.com). Diagram aktifitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Diagram aktifitas merupakan tahapan (*state*) diagram khusus, di mana sebagian besar tahapan adalah aksi dan sebagian besar transisi dipicu (*trigger*) oleh selesainya status sebelumnya (pemrosesan internal atau *internal processing*). Oleh karena itu diagram aktifitas tidak menggambarkan tingkah laku internal (*behaviour internal*) sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktifitas.

Sama seperti tahapan, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Keputusan (*decision*) digunakan untuk menggambarkan tingkah laku pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* atau transisi menyebar dan *join* atau transisi menyatu) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Diagram aktifitas dapat dibagi menjadi beberapa object garis batas (*swimlane*) untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Beberapa alasan membuat diagram aktifitas adalah sebagai berikut:









1. Menganalisis diagram *use case* lebih rinci,
2. Mengidentifikasi kondisi sebelum dan sesudah suatu *use case* (*pre and post condition*),
3. Menemukan *use case* baru yang tersembunyi.

Cara untuk membuat diagram aktifitas adalah:

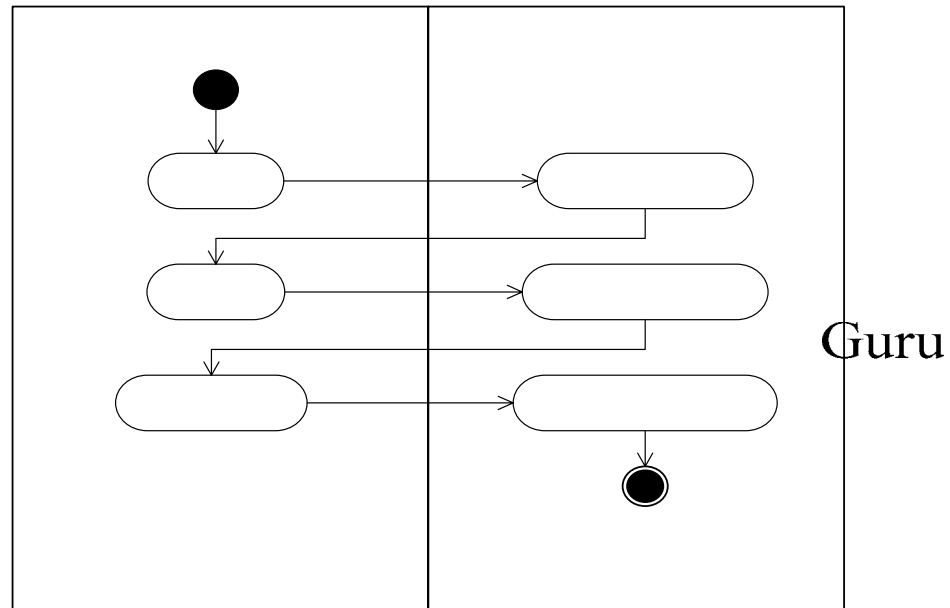
1. Identifikasi *use case*,
2. Pemodelan awal untuk setiap *use case*,
3. Pemodelan alternatif untuk setiap *use case*,
4. Menambahkan garis *swimlane*,
5. Refine aktivitas level akhir ke dalam diagram aktifitas.

Notasi diagram aktifitas dapat digambarkan pada Tabel 2.2 berikut:

Tabel 2.2 Notasi diagram aktifitas (Djajono, 2005: 85-93)

Notasi	Nama	Keterangan
	Action State	Kegiatan atau proses yang dilakukan.
	Initial state	Terminasi awal proses dilakukan.
	Final state	Terminasi akhir proses yang dilakukan.
	Transition (fork)	Transisi satu proses ke dua atau lebih proses berikutnya.
	Transition (join)	Transisi dua atau lebih proses ke satu proses berikutnya.
	Swimlane	Batasan proses yang dilakukan oleh aktor maupun oleh aplikasi.
	Control Flow	Alur kontrol dari satu proses ke proses lainnya.
	Decision	Percabangan proses pada kondisi tertentu.

Contoh dari diagram aktifitas dapat digambarkan pada Gambar 2.8 berikut:



Gambar 2.8 Contoh diagram aktifitas (Djajono, 2005: 88)

Dari Gambar 2.8, terdapat dua *swimlane* yaitu Guru dan antarmuka situs web. Guru akan melakukan logon pada antarmuka situs web. Antarmuka situs web akan melakukan validasi pengguna. Setelah validasi, antarmuka situs web akan menampilkan daftar pilihan siswa. Guru akan memilih salah satu siswa dan antarmuka situs web akan menampilkan info siswa. Guru melakukan perubahan pada info siswa dan antarmuka situs web memastikan info pengguna dan aktifitas siswa dengan disimpannya nilai (data) pada basis data.

2.7 Model Hubungan Entitas

Model hubungan entitas (*entity relationship model*) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa dunia nyata (*real world*) terdiri dari objek-objek dasar yang mempunyai hubungan atau relasi antar objek-objek tersebut (Marlinda, 2004: 17). Relasi antar

objek dilukiskan dengan menggunakan simbol-simbol grafis tertentu.

Model hubungan entitas adalah suatu penyajian data dengan menggunakan entitas (*entity*) dan hubungan (*relationship*). Diperkenalkan pada tahun 1976 oleh P.P. Chen. Alasan diperlukan model hubungan entitas:

1. Dapat menggambarkan hubungan antar entitas dengan jelas,
2. Dapat menggambarkan batasan jumlah entitas dan partisipasi antar entitas,
3. Mudah dimengerti oleh pemakai,
4. Mudah disajikan oleh perancang basis data.

Komponen-komponen yang terdapat di dalam model hubungan entitas yaitu:

1. Entitas

Entitas adalah sesuatu yang dapat dibedakan dalam dunia nyata di mana informasi yang berkaitan dengannya dikumpulkan. Set entitas (*entity set*) adalah kumpulan entitas yang sejenis. Simbol yang digunakan untuk entitas adalah persegi panjang. Set entitas dapat berupa:

- a. Entitas yang bersifat fisik, yaitu entitas yang dapat dilihat. Contohnya: rumah, kendaraan, mahasiswa, dosen, dan lain-lain,
- b. Entitas yang bersifat konsep atau logik, yaitu entitas yang tidak dapat dilihat.

Contohnya: pekerjaan, perusahaan, rencana, mata kuliah, dan lain-lain.

2. Hubungan

Hubungan adalah hubungan yang terjadi antara satu atau lebih entitas. Hubungan tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara entitas tersebut. Set hubungan (*relationship set*) adalah kumpulan hubungan yang sejenis. Simbol yang digunakan adalah bentuk belah ketupat atau persegi.

3. Atribut

Atribut adalah karakteristik dari entitas atau hubungan yang menyediakan penjelasan detail tentang entitas atau hubungan tersebut. Nilai atribut (*attribute value*) adalah suatu data aktual atau informasi yang disimpan di suatu atribut di dalam suatu entitas atau hubungan. Simbol yang digunakan adalah bentuk oval. Terdapat dua jenis atribut, yaitu:

- a. Kunci pengenal (*identifier key*), untuk menentukan suatu entitas secara unik,
- b. Atribut non kunci (*nonkey attribute, descriptor*), untuk menentukan karakteristik dari suatu entitas yang tidak unik.

4. Tipe indikator (*indicator type*)

Tipe indikator terdiri atas:

- a. Tipe indikator objek asosiatif (*indicator type associative object*), berfungsi sebagai objek dan suatu hubungan,
- b. Tipe inidakator super (*indicator type supertipe*), terdiri dari suatu objek dan satu subkategori atau lebih yang dihubungkan dengan satu hubungan yang tidak bernama.

5. Perbandingan kardinalitas

Perbandingan kardinalitas (*cardinality ratio*) menjelaskan hubungan batasan jumlah keterhubungan suatu entitas dengan entitas lainnya atau banyaknya entitas yang bersesuaian dengan entitas yang lain melalui suatu hubungan. Jenis Perbandingan kardinalitas:

- a. Satu ke satu atau *one to one* (1 : 1), yaitu hubungan satu entitas dengan satu entitas.

- b. Satu ke banyak atau *one to many* (1 : M) dan banyak ke satu atau *many to one* (M : 1), yaitu hubungan satu entitas dengan banyak entitas atau banyak entitas dengan satu entitas.
- c. Banyak ke banyak atau *many to many* (M : N), yaitu hubungan banyak entitas dengan banyak entitas.

6. Derajat hubungan

Derajat hubungan (*relationship degree*) menyatakan jumlah entitas yang berpartisipasi di dalam suatu hubungan. Derajat hubungan terdiri atas:

- a. Derajat satu (*unary degree*) adalah derajat yang memiliki satu hubungan untuk satu entitas,
- b. Derajat dua (*binary degree*) adalah derajat yang memiliki satu hubungan untuk dua entitas,
- c. Derajat tiga (*ternary degree*) adalah derajat yang memiliki satu hubungan untuk tiga atau lebih entitas.

7. Batasan keikutsertaan

Batasan keikutsertaan (*participation constraint*) menjelaskan apakah keberadaan suatu entitas tergantung pada hubungan dengan entitas lain. Terdapat dua macam batasan keikutsertaan, yaitu:

- a. Partisipasi total (*total participation*), yaitu keberadaan suatu entitas tergantung pada hubungannya dengan entitas lain. Di dalam diagram hubungan entitas (*entity relationship diagram*) digambarkan dengan dua garis penghubung antar entitas dan hubungan.

- b. Partisipasi sebagian (*partial participation*), yaitu keberadaan suatu entitas tidak tergantung pada hubungan dengan entitas lain. Di dalam diagram hubungan entitas digambarkan dengan satu garis penghubung.


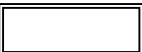
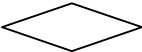



8. Representasi dari set entitas

Set entitas (*entity set*) direpresentasikan dalam bentuk tabel dan nama yang unik. Setiap tabel terdiri dari sejumlah kolom. Masing-masing kolom diberi nama yang unik pula. Set entitas *set* terbagi atas:

- a. Set entitas kuat (*strong entity set*), adalah set entitas yang satu atau lebih atributnya digunakan oleh set entitas lain sebagai kunci (*key*), digambarkan dengan empat persegi panjang.
- b. Set entitas lemah (*weak entity set*), adalah set entitas yang tergantung (*dependent*) terhadap set entitas kuat. Keberadaan entitas tersebut tergantung keberadaan entitas lain. Entitas lain tersebut disebut pemilik identifikasi (*identifying owner*) dan hubungannya disebut hubungan identifikasi (*identifying relationship*). Entitas lemah selalu mempunyai partisipasi total dengan pemilik identifikasi. Entitas lemah digambarkan dengan persegi panjang bertumpuk.

Notasi model hubungan entitas dijelaskan pada Tabel 2.3.

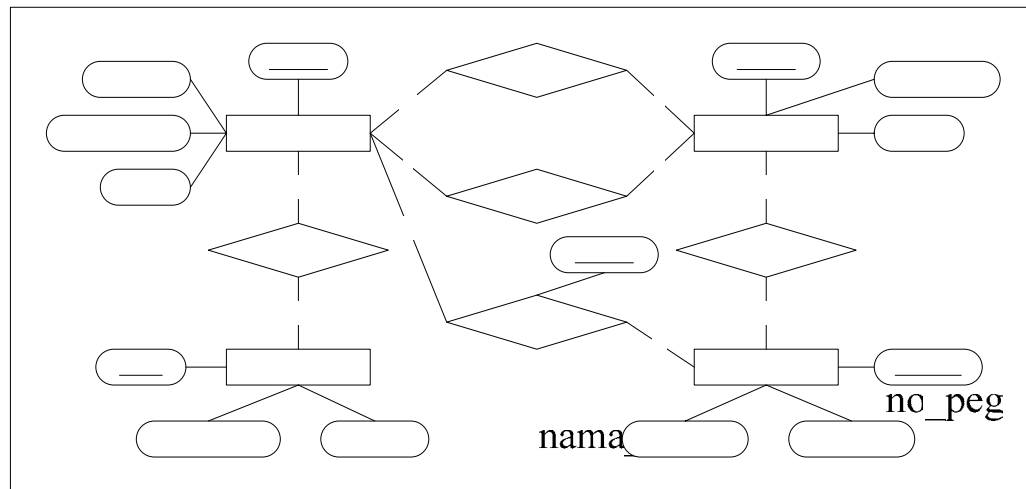
Tabel 2.3 Notasi model hubungan entitas (Marlinda, 2004: 17-22)

Notasi	Nama	Keterangan
	Set entity kuat	Set entitas yang satu atau lebih atributnya digunakan oleh set entitas lain sebagai kunci.
	Set entitas lemah	Set entitas yang tergantung terhadap set entitas kuat, keberadaan entitas tersebut tergantung keberadaan entitas lain.
	Hubungan	Hubungan terjadi antara satu atau lebih entitas.
	Atribut	Karakteristik dari entitas atau hubungan yang menyediakan penjelasan detail tentang entitas atau hubungan tersebut.
	Partisipasi total	Keberadaan suatu entitas tergantung pada hubungannya dengan entitas lain.
	Partisipasi sebagian	Keberadaan suatu entitas tidak tergantung pada hubungan dengan entitas lain.

Langkah-langkah membuat diagram hubungan entitas:

1. Tentukan entitas-entitas yang diperlukan,
2. Tentukan hubungan antar entitas-entitas,
3. Tentukan rasio kepentingan dan *participation constraint*,
4. Tentukan atribut-atribut yang diperlukan dari tiap entitas,
5. Tentukan kunci di antara atribut-atribut,
6. Hindari penamaan entitas, hubungan, dan atribut yang sama.

Contoh dari diagram hubungan entitas digambarkan pada Gambar 2.9 berikut.



Gambar 2.9 Contoh diagram hubungan entitas (Marlinda, 2004: 24)

Dari Gambar 2.9 terdapat empat buah entitas yaitu Pegawai, Departemen, Proyek, dan Tanggungan. Entitas Pegawai terdapat empat buah atribut yaitu no_peg, nama_peg, jenis_kelamin, dan gaji. *Primary key* dari entitas Pegawai adalah no_peg. Entitas Departemen memiliki tiga atribut yaitu no_dep, nama_dep, dan lokasi. *Primary key* dari entitas Departemen adalah no_dep. Entitas Proyek terdiri atas atribut no_pryk, nama_pryk, dan lokasi_pryk. *Primary key* entitas Proyek adalah no_pryk. Entitas keempat adalah entitas Tanggungan yang terdiri dari atribut nama, jenis_kelamin, dan hubungan dengan nama sebagai *primary key* entitas ini.

Hubungan dari empat buah entitas tersebut adalah sebagai berikut:

- Banyak pegawai menjadi anggota satu departemen (M : 1),
- Satu pegawai memimipin satu departemen (1 : 1),
- Banyak pegawai bekerja pada banyak proyek (M : N),
- Satu departemen membawahi banyak proyek (1 : M), dan
- Satu pegawai mempunyai banyak tanggungan (1 : M).

2.8 PHP (PHP *Hypertext Preprocessor*)

PHP adalah teknologi yang diperkenalkan tahun 1994 oleh *Rasmus Lerdorf*. Beberapa versi awal yang tidak di publikasikan digunakan pada situs pribadinya untuk mencatat siapa saja yang mengakses daftar riwayat hidup onlinenya. Versi pertama digunakan oleh pihak lain tahun 1995 dan terkenal sebagai *personal home page tools*.

PHP adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi server. Sebagian besar perintahnya berasal dari C, Java dan perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (*Common Gateway Interface*).

PHP bisa berinteraksi dengan hampir semua teknologi web yang sudah ada. Web developer bisa menulis sebuah program PHP yang mengeksekusi suatu program CGI di server web lain.

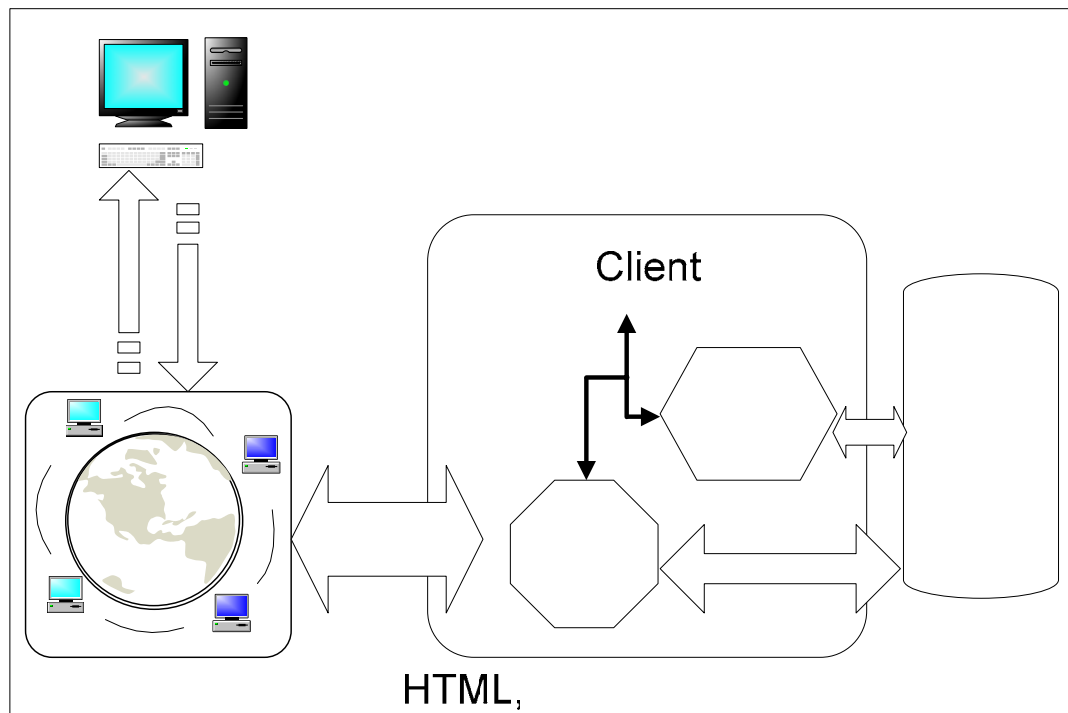
Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih bisa mempergunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan CGI, ISAP, atau dengan script seperti Perl, Awk atau Python selama proses migrasi ke aplikasi baru yang dibuat dengan PHP ini mempermudah dan memperluas peralihan antara teknologi lama dan teknologi baru.

PHP adalah *server-side programming* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Pada *server-side programming* perintah-perintah program dijalankan di *web server*, sedangkan pada *client-side programming* perintah di jalankan di *client*. Ketika pemakai internet akan membuka

suatu situs yang menggunakan *server-side programming* PHP, maka *server* akan memproses semua perintah PHP terlebih dulu kemudian mengirimkan hasilnya dalam format HTML ke *web browser* pemakai tersebut. Dengan demikian seorang pemakai internet tidak dapat melihat program yang ditulis dalam PHP sehingga keamanan dari halaman *web* menjadi lebih terjamin.

2.8.1 Arsitektur PHP dengan Basis Data

Skema dalam pembuatan isi situs yang dinamis antara HTML, PHP, dan basis data terdiri dari tiga tingkatan/tier (lihat Gambar 2.6). Pada Gambar 2.6, terdapat 2 komunikasi, satu dirancang untuk dapat diakses oleh manusia melalui web browser, dan lainnya untuk pertukaran data aplikasi-ke-aplikasi. Antarmuka yang mengandung bentuk HTML pada komputer client mengizinkan pengguna untuk memasuki SQL query. Bahasa PHP digunakan untuk membangun pernyataan SQL dan mengaitkan basis data, data lalu dikeluarkan dari basis data, dan pengguna disajikan oleh tampilan struktur tabel.



Gambar 2.10 Konsep kerja PHP dan Mysql

Tingkatan-tingkatan pada situs dinamis dijelaskan seperti di bawah ini:

1. Tingkatan Pertama

Tingkatan ini terdiri dari antarmuka, diterapkan melalui bentuk HTML dengan dan tanpa dukungan dari JavaScript. Hasilnya dikembalikan dan ditampilkan sebagai tabel. Ada juga antarmuka yang interaktif menggunakan frame, bentuk HTML, JavaScript, dan Java applet.

2. Tingkatan Kedua

Tingkatan Kedua terdiri dari web server Apache. Modul PHP mendukung untuk beberapa basis data backend (mSQL, MySQL and PostgreSQL), sebaik dengan pembangkitan Gambar GIF, dan pengelolaan data XML. Bahasa PHP memproses form masukan, membangkitkan SQL query, melakukan pencarian dan mengelola keluaran ke dalam dokumen HTML.

WWW
First Tier

HTTP
Java RMI

3. Tingkatan Ketiga

Tingkatan ketiga terdiri dari mesin basis data itu sendiri. Di sini, digunakan basis data MySQL.

2.8.2 Aplikasi PHP dengan MySQL

Dalam pembuatan situs kebanyakan PHP direkankan dengan basis data MySQL. PHP menyediakan script untuk melakukan manipulasi isi ke basis data MySQL dengan sintaks di bawah ini:

```
mysql_connect("localhost","username","password")
mysql_db_query("namadbase","manipulasi")
```

Pada baris pertama, username dan password diambil berdasarkan username dan password di MySQL. Sedangkan di baris kedua **namadbase** merupakan nama database di MySQL dan **manipulasi** diisi dengan sintaks dasar SQL seperti select, update, dan delete.

Contoh: sintaks dasar pada halaman **agenda.php**:

a. insert:

```
mysql_db_query($dbname,"insert into $tb_agenda(tanggal,judul,isi) values('$tgl','$judul','$isi')");
```

b. update:

```
mysql_db_query($dbname,"update $tb_agenda set tanggal='$tgl', judul='$judul', isi='$isi' where agid='$agid'");
```

c. delete:

```
mysql_db_query($dbname,"delete from $tb_agenda where agid='$agid'");
```


BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

Keamanan data pada PT. Lelco Trindo Graha Nusantara merupakan masalah yang penting, karena bisa saja terjadi pengambilan data oleh pihak lain dengan cara-cara tertentu, misalnya menggandakan data. Apabila data tidak dimanipulasi, maka isi data dapat langsung dibaca oleh pihak lain, apabila data yang bersifat penting atau rahasia. Oleh karena itu, dikembangkan metode enkripsi yang memungkinkan orang lain tidak bisa membuka data tanpa seizin pemiliknya. Metode yang dipakai, yaitu adalah MD-5. Dengan menggunakan metode ini, maka lapisan keamanan data pun menjadi sulit untuk dibongkar, karena data dilindungi dengan pengamanan.

Perkembangan komputer yang begitu pesat dengan teknik komputer yang terus berkembang, menyebabkan keamanan data harus lebih ditingkatkan. Keamanan data dalam hal ini lebih meliputi kerahasiaan (*confidentiality*), keutuhan (*integrity*), dan keaslian (*authenticity*) data. Kerahasiaan data merupakan keamanan isi dari data yang tidak boleh diketahui oleh pihak lain. Keutuhan data, merupakan keamanan dalam hal ukuran dan isi data. Keaslian data, merupakan keamanan dalam hal kepemilikan data dan termasuk disini tanggal dan jam (*timestamp*) modifikasi data. Jadi permasalahan timbul apabila hal-hal tersebut dimanipulasi oleh pihak lain.

Pengamanan data terhadap pihak lain, dapat dilakukan dengan backup data pada penyimpanan data berupa disket ataupun *hard disk*. Akan tetapi usaha yang dilakukan ini masih dianggap beresiko karena pihak lain masih dapat mengambil data tersebut. Bila data tidak diberikan sistem keamanan berdasarkan hak akses pengguna, maka isi dari data yang penting bisa langsung dibaca oleh pihak lain yang tidak berkepentingan.

Dari permasalahan tersebut didapatkan proses yang terdapat pada tingkat administrator adalah proses keabsahan, proses pembatasan informasi, dan proses pengolahan data. Berikut penjabaran dari ketiga proses tersebut:

1. Proses keabsahan

Proses ini untuk memastikan kebenaran pengaksesan pengguna. Masukannya terdiri dari *username*, *password*, dan status pengguna. Keluaran dari proses ini adalah nilai boolean, yaitu benar (*true*) atau salah (*false*). Jika *true* akan masuk ke halaman administrator. Jika *false* maka akan muncul teks peringatan.

2. Proses pembatasan informasi

Pada proses ini terdapat data level administrator atau sales sesuai dengan tingkatannya. Pada saat pengguna memasukkan nama dan *password* maka aplikasi akan secara langsung mengenali level pengguna maka informasi yang dapat diakses oleh pengguna akan dibatasi sesuai dengan level akses yang dimilikinya.

3. Proses pengolahan data.

Proses ini dijalankan ketika pengguna telah memasukkan data ke dalam basis data.

Pada aplikasi sistem keamanan ini juga terdapat permasalahan terhadap keamanan masuk (*login*) data *username* dan *password*. Nilai data-data berupa *username* dan *password* baik di tingkat administrator dan sales disimpan menggunakan fungsi MD-5 di penyedia (*server*). Jika nilai data tersebut dapat dibaca maka akan mudah untuk mengakses dengan *username* dan *password* pengguna lain. Untuk itu penulis membuat enkripsi pada nilai data yang disimpan menggunakan fungsi MD-5.

Dalam melakukan pengujian harus ada kebutuhan sistem yang didasarkan pada sudut pandang pemakai, yaitu:

- a. Terdapat bantuan penggunaan.
- b. Nama hubungan (*link*) sesuai dengan proses yang dikerjakan.
- c. Tampilan setiap objek pada setiap halaman harus sesuai dengan fungsi halamannya.
- d. Jumlah pengguna dan produk dapat bertambah.

Penulis menggunakan lima tabel pada aplikasi keamanan, tabel-tabel tersebut adalah:

1. Tabel login administrator
2. Tabel login buku tamu
3. Tabel produk
4. Tabel report
5. Tabel tipeuser.

3.1.1 Rekayasa Sistem

PT. Lelco Trindo Graha Nusantara merupakan perusahaan yang bergerak dibidang produksi, distribusi dan penjualan produk lampu. Dalam manajemen data yang terdapat pada perusahaan tersebut terdapat lima level inti untuk mengkoordinir semua lalu-lintas data di dalam perusahaan yaitu :

1. Super Administrator

Super administrator adalah level tertinggi dalam perusahaan yang mempunyai akses keseluruhan data dalam perusahaan. Dalam hal ini, super admin memiliki hak untuk menghapus, menambah, mengubah, dan melihat semua pengakses dalam *database* perusahaan. Super administrator bertanggung jawab langsung dengan pimpinan perusahaan dan mempunyai posisi di *top management*.

2. Top Administrator

Top administrator berada pada level kedua setelah super administrator dengan hak akses yang hampir sama dengan super admin tetapi mempunyai pembatasan pada hak akses pada level yang sama dan level di atasnya. Top administrator berfungsi sebagai perwakilan dari super administrator dalam mengatur lalu-lintas data yang masuk ke perusahaan.

3. Product Administrator

Product administrator bertanggung jawab atas produk yang ada di perusahaan.

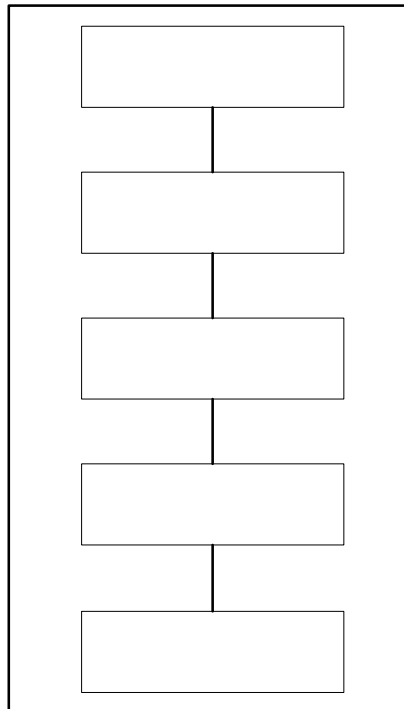
4. Sales Administrator

Sales administrator bertugas untuk mengolah laporan yang di terima dari sales perusahaan.

5. Sales

Sales adalah tenaga penjualan yang dimiliki perusahaan untuk memasarkan produk.

Aplikasi keamanan data menggunakan *user access* ini berfungsi sebagai pengamanan terhadap data yang ada dalam perusahaan agar hanya dapat diakses berdasarkan level seperti yang berada pada Gambar 3.1.



Gambar 3.1. Level *user access*

3.2 Perancangan

Berdasarkan analisis yang telah dibuat, penulis menuangkannya ke dalam bentuk yang mudah dimengerti. Pembuatan diagram alir data untuk data yang bergantung kepada proses beserta kamus datanya, tinjauan diagram keterhubungan entitas terhadap aplikasi keamanan, rancangan tabel, dan rancangan antar muka, juga

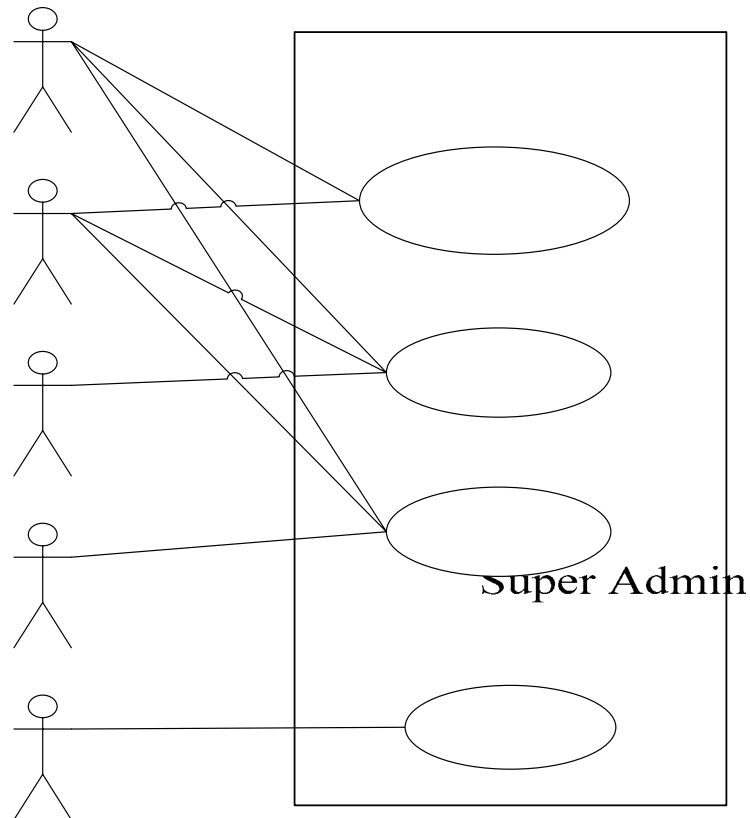
diperlukan suatu cara dengan aturan-aturan tertentu yang ditetapkan, agar data tetap terjaga dalam hal kerahasiaan, keutuhan dan keasliannya. Aturan-aturan yang ditetapkan tersebut, dimaksudkan agar data yang dimanipulasi tetap terjaga keamanannya. Dalam arti, data dapat dikembalikan ke bentuk asli, dengan kondisi seperti semula.

Banyak cara untuk mengamankan data, seperti menggunakan *password* dalam pembacaan data, proteksi terhadap data, *backup* data, dan lain sebagainya. Cara yang dilakukan disini adalah memanipulasi data sedemikian rupa, sehingga isi dari data asli diubah menjadi bentuk lain. Bentuk lain adalah bentuk yang tidak dimengerti oleh pihak lain, sehingga untuk mengubah data asli diperlukan suatu proses. Pemanipulasian data dilakukan dengan cara mengubah isi data yang berupa karakter-karakter menjadi karakter-karakter yang berbeda. Karakter-karakter disini berdasarkan karakter-karakter ASCII yang merupakan karakter-karakter standar yang digunakan oleh komputer. Untuk melakukan proses pengubahan ini, dapat digunakan suatu kunci sebagai dasar untuk mengubah karakter-karakter yang terdapat pada data tersebut. Dapat juga dilakukan dengan mengacak atau mengubah posisi karakter-karakter ke posisi lain. Setelah isi data asli diubah, juga diperlukan suatu proses untuk mengembalikan ke data asli.

3.2.1 Diagram Use Case

Aplikasi ini dikembangkan oleh PT. Lelco Trindo Graha Nusantara untuk menjaga keamanan data perusahaan dari orang-orang yang tidak mempunyai hak untuk melihat atau mengambil data tersebut. Keamanan data itu sendiri akan dipantau langsung oleh super admin yang ditunjuk perusahaan sebagai pengelola

aplikasi dan pemberi hak akses. Diagram *use case* dapat dijelaskan pada Gambar 3.2.



Gambar 3.2 Diagram *use case* aplikasi
Top Admin

Keterangan dari diagram use case pada Gambar 3.2 adalah sebagai berikut:

1. Nama *use case*: Memanajemen data admin

Aktor: Super Admin, Top admin

Deskripsi: Kegiatan untuk menambah, menghapus, merubah data admin pada perusahaan.

Produk Admin

Exception (pengecualian): Top admin tidak dapat merubah data administrator di level yang sama dan diatas top admin. Data tidak akan disimpan pada basis data jika seluruh data tidak diisi dengan lengkap.

Sales Admin

Skenario:

1. Admin memilih menu tambah admin, menu edit atau menu hapus,
2. Aplikasi menampilkan antarmuka tambah admin sesuai dengan sub menu yang dipilih,
3. Admin melakukan proses masukan data,
4. Pengguna menekan tombol input untuk menyimpan data pada basis data,
5. Aplikasi menyimpan data yang dimasukkan pada basis data,
6. Aplikasi kembali ke menu administrator menyimpan data yang dimasukkan pada basis data.

Alternatif: Nomor 5, jika input data gagal, aplikasi menampilkan kotak dialog masukan data gagal, admin akan kembali ke menu administrator

Precondition (kondisi awal): Aplikasi akan menampilkan antarmuka tambah admin atau edit admin.

Postcondition (kondisi akhir): Data yang dimasukkan telah tersimpan pada basis data.

2. Nama *use case*: Memanajemen data produk

Aktor: Super Admin, Top admin, produk admin.

Deskripsi: Kegiatan untuk menambah, menghapus, merubah data produk pada perusahaan.

Exception (pengecualian): Data tidak akan disimpan pada basis data jika seluruh data tidak diisi dengan lengkap.

Skenario:

1. Admin memilih tambah barang, menu edit atau menu hapus,
2. Aplikasi menampilkan antarmuka tambah barang sesuai dengan sub menu yang dipilih,
3. Admin melakukan proses masukan data,
4. Admin menekan tombol input untuk menyimpan data pada basis data,
5. Aplikasi menyimpan data yang dimasukkan pada basis data,
6. Aplikasi kembali ke menu Update barang menyimpan data yang dimasukkan pada basis data.

Alternatif: Nomor 5, jika input data gagal, aplikasi menampilkan kotak dialog masukan data gagal, admin akan kembali ke menu administrator

Precondition (kondisi awal): Aplikasi akan menampilkan antarmuka update barang.

Postcondition (kondisi akhir): Data yang dimasukkan telah tersimpan pada basis data.

3. Nama *use case*: Memanajemen sales

Aktor: Super Admin, Top admin, sales admin.

Deskripsi: Kegiatan untuk menambah, menghapus, merubah data sales pada perusahaan.

Exception (pengecualian): Data tidak akan disimpan pada basis data jika seluruh data tidak diisi dengan lengkap.

Skenario:

1. Admin memilih menu update report,
2. Aplikasi menampilkan antarmuka update report sesuai dengan sub menu yang dipilih,
3. Admin melakukan proses masukan data report yang dimasukkan oleh sales,
4. Admin menekan tombol compile untuk membuat rangkuman report yang ada pada basis data,
5. Admin menekan tombol input untuk menyimpan rangkuman data report pada basis data,
6. Aplikasi menyimpan data yang dimasukkan pada basis data,
7. Aplikasi kembali ke menu Update report menyimpan data yang dimasukkan pada basis data,

Alternatif: Nomor 6, jika input data gagal, aplikasi menampilkan kotak dialog masukan data gagal, admin akan kembali ke menu administrator

Precondition (kondisi awal): Aplikasi akan menampilkan antarmuka update sales.

Postcondition (kondisi akhir): Data yang dimasukkan telah tersimpan pada basis data.

4. Nama *use case*: Masukkan data penjualan

Aktor: Sales.

Deskripsi: Kegiatan untuk memasukkan data laporan penjualan yang didapat oleh sales pada perusahaan.

Exception (pengecualian): Data tidak akan disimpan pada basis data jika seluruh data tidak diisi dengan lengkap.

Skenario:

1. Sales memilih menu submit laporan,
2. Aplikasi menampilkan antarmuka submit laporan sesuai dengan sub menu yang dipilih,
3. Sales melakukan proses masukan data report,
4. Sales menekan tombol submit untuk menyimpan data report pada basis data,
5. Aplikasi menyimpan data yang dimasukkan pada basis data,
6. Aplikasi kembali ke menu submit report menyimpan data yang dimasukkan pada basis data.

Alternatif: Nomor 4, jika input data gagal, aplikasi menampilkan kotak dialog masukan data gagal, admin akan kembali ke menu administrator

Precondition (kondisi awal): Aplikasi akan menampilkan antarmuka sales.

Postcondition (kondisi akhir): Data yang dimasukkan telah tersimpan pada basis data.

3.2.2 Basis Data

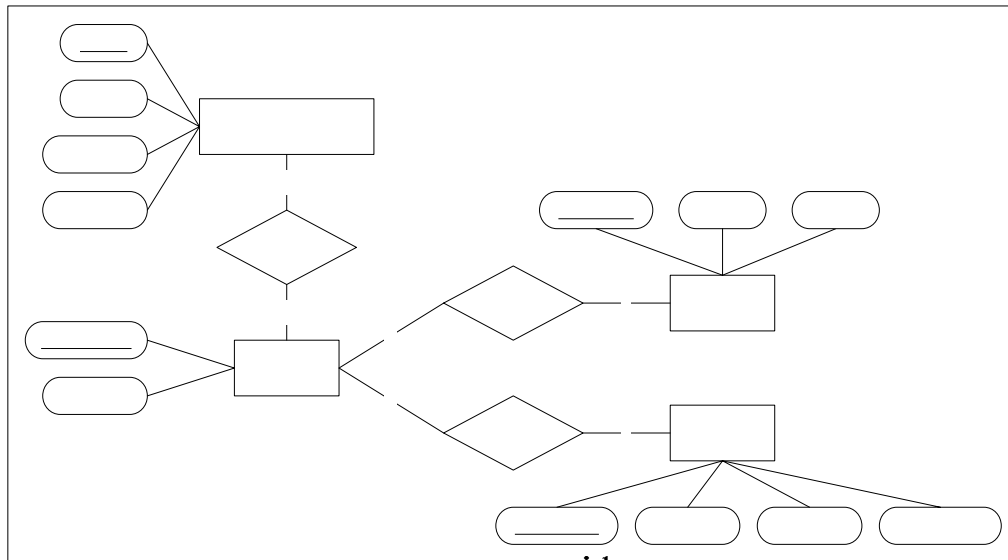
Analisis pada segi basis data aplikasi yang dikembangkan akan ditinjau dari diagram hubungan entitas dan struktur entitas dari basis data.

3.2.2.1 Diagram Hubungan Entitas

Aplikasi yang dikembangkan terdiri atas empat tabel yang saling berhubungan, yaitu tabel login, tipeuser, report, dan produk. Empat tabel ini memiliki hubungan kardinalitas sebagai berikut:

1. Satu login memiliki satu tipeuser,
2. Banyak tipeuser mengisi banyak report, dan
3. Banyak tipeuser mengisi banyak produk.

Diagram hubungan entitas dari aplikasi digambarkan pada Gambar 3.3.



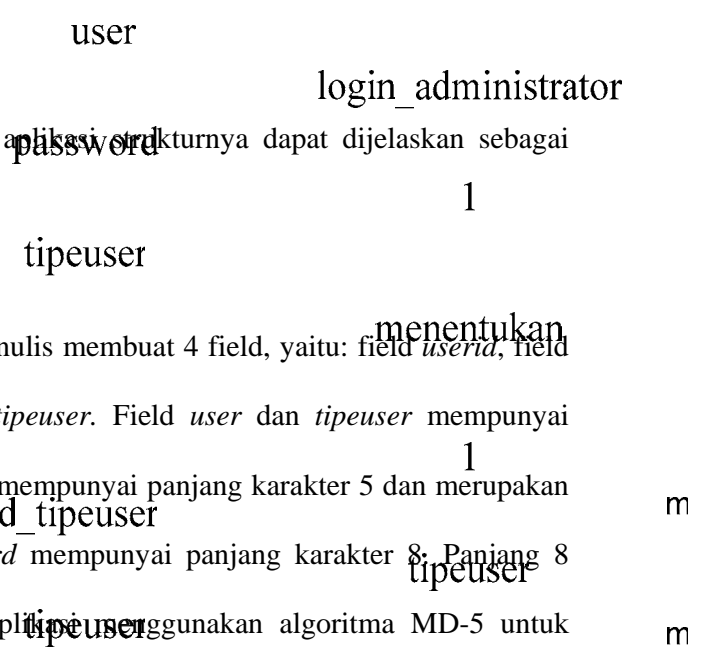
Gambar 3.3 Diagram hubungan entitas aplikasi

3.2.3 Struktur Basis Data

Empat tabel pada basis data aplikasi strukturnya dapat dijelaskan sebagai berikut:

1. Tabel login administrator

Pada tabel *login administrator*, penulis membuat 4 field, yaitu: field *userid*, field *user*, field *password*, dan field *tipeuser*. Field *user* dan *tipeuser* mempunyai panjang karakter 20, Field *userid* mempunyai panjang karakter 5 dan merupakan *primary key*, dan Field *password* mempunyai panjang karakter 8. Panjang 8 karakter ini akan diubah oleh aplikasi menggunakan algoritma MD-5 untuk menyulitkan orang-orang yang ingin membongkar kombinasi *password*. Panjang



8 karakter diambil berdasarkan kata-kata yang dimengerti oleh penulis. Tabel login administrator dijelaskan pada Tabel 3.1.

Tabel 3.1 Entitas login administrator

Atribut	Tipe	Panjang	Keterangan
Userid	Char	5	Nomor id user
User	Varchar	20	Nama user
password	Varchar	8	Password (MD-5)
Tipe user	Varchar	20	Model user

2. Tabel tipeuser

Pada tabel data tipeuser, penulis membuat 2 field, yaitu: field kd_tipeuser dan field tipeuser, Field-field ini digunakan untuk menampilkan level pengguna ketika pemilih sukses *login*. Field id akan menjadi *primary key* yang menjadi dasar level pengguna pada aplikasi ini. Panjang field id adalah 3 karakter. Field tipeuser mempunyai panjang 20 karakter dimana jenis level pengguna tidak melebihi 20 karakter. Penjelasan panjang nama sudah terdapat pada tabel produk. Tabel tipeuser dijelaskan pada tabel 3.2.

Tabel 3.2 Entitas tipeuser

Atribut	Tipe	Panjang	Keterangan
Kd_tipeuser	char	3	Nomor kode tipeuser
tipeuser	varchar	20	Jenis user

3. Tabel produk

Pada tabel produk, penulis membuat 4 field, yaitu: field kd_produk, field nama, field gambar dan field keterangan. Field kd_produk adalah kode produk yang ada di perusahaan mempunyai 3 karakter dan merupakan *primary key*. Panjang 3

karakter ini digunakan karena jumlah produk yang ada diperkirakan mencapai ratusan macam. Field nama mempunyai panjang 20 karakter. Nama terpanjang yang menjadi dasar penulis adalah Eka Adi Rahayu yang mempunyai 14 karakter sedangkan 6 karakter sisa merupakan pengalokasian karakter tambahan bila ternyata ada yang lebih dari 14 karakter. Field gambar mempunyai panjang 30. Panjang 30 karakter ini di gunakan untuk memasukkan gambar dan nama produk kedalam database. Field keterangan menggunakan tipe text. Tipe text digunakan untuk memberikan keterangan tentang produk yang di tampilkan. Tabel produk dijelaskan pada tabel 3.3.

Tabel 3.3 Entitas produk

Atribut	Tipe	Panjang	Keterangan
kd_produk	Char	10	Nomor kode produk
Nama	Char	10	Nama produk
Gambar	Char	10	Gambar produk
Keterangan	Char	10	Keterangan-keterangan gambar produk

4. Tabel report

Pada tabel report, penulis membuat 3 field, yaitu: field kd_report, field nama, dan field file. Field kd_report mempunyai panjang 3 karakter karena jumlah sales yang ada pada perusahaan bisa mencapai angka ratusan dan merupakan *primary key*. Field nama panjang karakter 20 yang berisi nama sales. Panjang nama sendiri tidak lebih dari 20 karakter. Penjelasan panjang nama sudah terdapat pada tabel produk. Field file mempunyai panjang 50 karakter yang berisi laporan yang dimasukkan oleh sales. Panjang 50 karakter didapat dengan memperkirakan bahwa laporan yang harus diisi setiap sales mempunyai 50 karakter.

Tabel report dijelaskan pada tabel 3.4.

Tabel 3.4 Entitas report

Atribut	Tipe	Panjang	Keterangan
kd_report	Char	3	Nomor kode report
Nama	Char	20	Nama/judul report
File	Char	50	File berupa laporan yang diisi oleh sales

3.2.4 Proses

Analisis pada segi proses yang berjalan pada aplikasi yang dikembangkan akan ditinjau dari diagram aktifitas.

3.2.4.1 Diagram aktifitas

Diagram aktifitas menjelaskan tentang berbagai aktifitas yang berlangsung selama pengguna berinteraksi dengan aplikasi. Diagram aktifitas menjelaskan diagram *use case* secara lebih rinci. Pada diagram *use case* terdapat lima proses. Proses-proses tersebut akan dijelaskan secara lebih rinci sebagai berikut:

1. Login

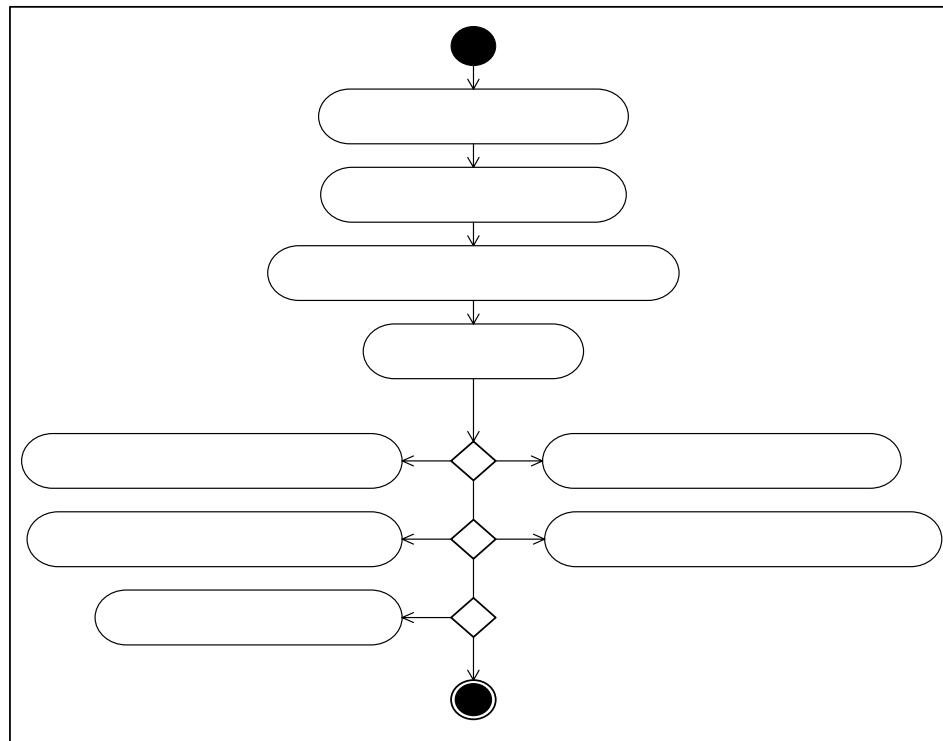
Login adalah sebuah proses awal yang dilakukan oleh pengguna (dalam hal ini adalah administrator dan sales) untuk mengakses ke database perusahaan. Proses ini berfungsi untuk membatasi akses pada aplikasi, sehingga hanya pengguna yang berwenang saja yang dapat mempergunakannya. Pengguna akan memasukkan data berupa nama pengguna dan kata kunci. Pengguna selanjutnya akan menekan tombol Masuk. Tombol ini akan membuat aplikasi melakukan validasi nama pengguna dan kata kunci yang dimasukkan apakah sesuai dengan data nama pengguna dan data kata kunci yang ada pada basis data aplikasi. Jika

data nama pengguna dan data kata kunci sesuai, maka aplikasi secara otomatis akan menentukan tipe user dan menampilkan antarmuka atau layar menu administrator.

Pada antarmuka login, pengguna ada lima antarmuka sesuai dengan tipe user yang menggunakan aplikasi, yaitu:

- a. Halaman super admin, halaman yang hanya dapat diakses oleh super admin (aplikasi ini menggunakan nama indri sebagai super admin).
- b. Halaman top admin, halaman yang hanya dapat diakses oleh top admin dan super admin dan dapat melakukan perubahan atau pengeditan terhadap data tersebut.
- c. Halaman produk admin, halaman yang hanya dapat diakses oleh produk admin dan level admin di atasnya (super admin dan top admin) halaman untuk memproses seluruh data produk.
- d. Halaman admin sales, halaman yang hanya dapat diakses oleh sales admin dan level admin di atasnya (super admin dan top admin) menu untuk dan memproses data report yang dimasukkan oleh sales.
- e. Halaman sales, halaman yang diakses oleh sales untuk memasukkan report.

Diagram aktifitas proses login administrator digambarkan pada Gambar 3.4.



Gambar 3.4 Diagram aktifitas login administrator

2. Halaman update administrator

Pengguna yang telah melakukan proses login dan telah dikenali oleh aplikasi sebagai super admin dan top admin dapat melakukan proses data admin dengan memilih menu update admin. Data yang dapat diproses dapat dikelompokkan menjadi tiga kelompok, yaitu:

- a. Menu tambah admin, yaitu submenu untuk menambah data administrator.
- b. Menu rubah admin, yaitu submenu untuk mengubah data administrator yang sudah ada dalam basis data administrator.
- c. Menu hapus admin, yaitu menu untuk menghapus data administrator yang ada dalam basis data.

Memasuk

memasuk

verifikasi data

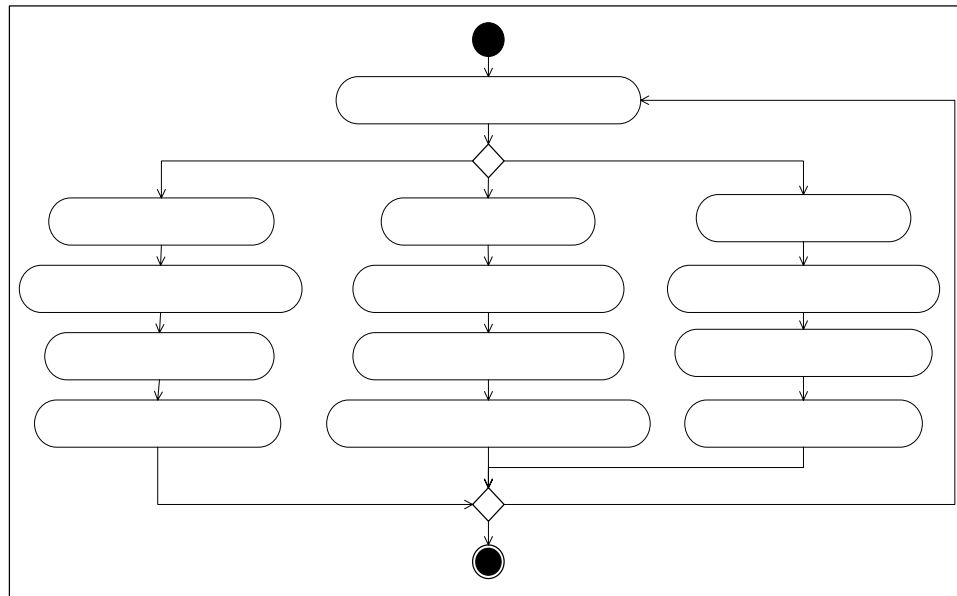
pener

menampilkan halaman super admin

menampilkan halaman sales admin

menampilkan halaman sales

Diagram aktifitas proses update administrator digambarkan pada Gambar 3.5.



Gambar 3.5 Diagram aktifitas update administrator

3. Halaman produk admin

memilih menu tambah admin

memil

Halaman produk admin dapat diakses pengguna yang telah melakukan proses login dan telah dikenali oleh aplikasi sebagai admin. Pengguna dapat melakukan proses data produk dengan memilih menu update produk. Data yang dapat diproses dapat dikelompokkan menjadi tiga kelompok, yaitu:

menampil

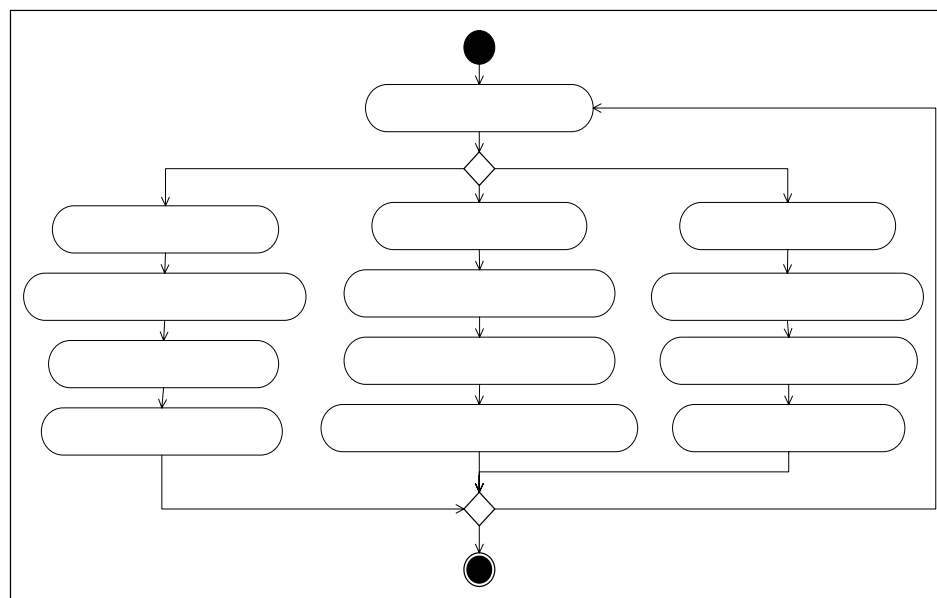
memasukk

- Menu tambah produk, yaitu submenu untuk menambah data produk baru.
- Menu rubah produk, yaitu submenu untuk merubah data produk yang sudah ada dalam basis data produk.
- Menu hapus produk, yaitu menu untuk menghapus data produk yang ada dalam basis data.

menyimpan da

Pengguna akan memilih salah satu dari submenu dan aplikasi akan menampilkan data sesuai dengan submenu yang dipilih. Pengguna memilih data yang akan dirubah. Setelah memilih, pengguna akan merubah data yang

mengalami perubahan ataupun kesalahan. Setelah selesai melakukan perubahan, aplikasi akan mengecek apakah data yang dirubah telah terisi dengan lengkap. Apabila data telah lengkap maka aplikasi akan menyimpan perubahan data pada basis data dan pengguna akan kembali ke menu utama untuk melakukan proses lainnya. Jika data tidak lengkap, maka pengguna harus melakukan ulang proses perubahan data. Diagram aktifitas proses manajemen data produk digambarkan pada Gambar 3.6.



Gambar 3.6 Diagram aktifitas manajemen data produk

4. Halaman sales admin

Halaman sales admin dapat diakses pengguna yang telah melakukan proses login dan telah dikenali oleh aplikasi sebagai sales admin yang digunakan untuk memproses data yang telah masukkan oleh sales. Pemrosesan data dapat digambarkan sebagai berikut :

- a. Halaman report sales, yaitu halaman untuk melihat data report yang telah dimasukkan oleh sales.

memilih menu tambah produk

menampilkan halaman tambah produk

memasukkan data produk baru

menam

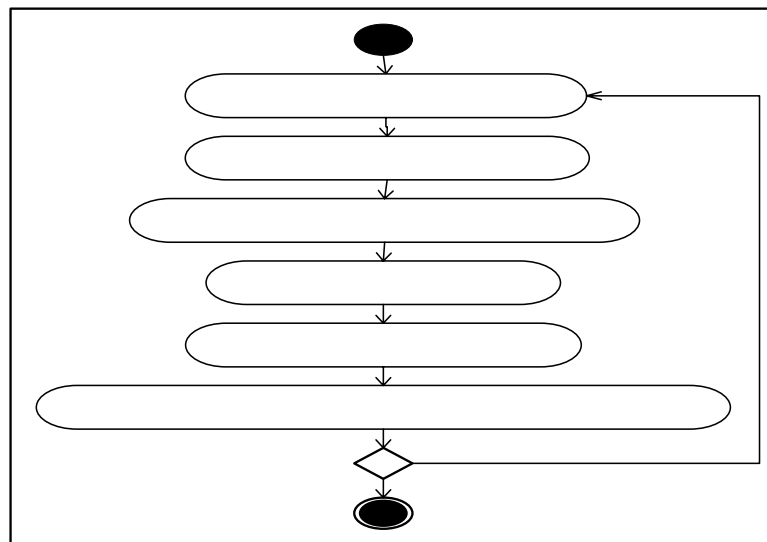
memil

menampil

memasuk

- b. Menu compile, yaitu submenu untuk membuat rangkuman data yang telah dimasukkan oleh sales.
- c. Menu submit data, yaitu submenu untuk menyimpan rangkuman data sales kedalam basis data.

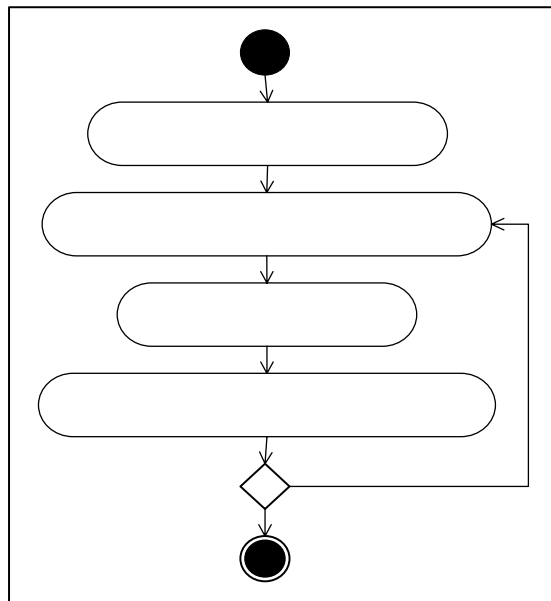
Sebelum membuat rangkuman data sales, sales admin sales akan merubah data yang mengalami perubahan ataupun kesalahan. Setelah selesai melakukan perubahan, aplikasi akan mengecek apakah data yang dimasukkan oleh sales telah terisi dengan lengkap. Apabila data telah lengkap maka sales admin dapat membuat rangkuman data sales lalu aplikasi akan menyimpan rangkuman data sales pada basis data dan sales admin akan kembali ke menu utama untuk melakukan proses lainnya. Jika data tidak lengkap, maka sales admin harus melakukan ulang proses perubahan data. Diagram aktifitas proses manajemen data sales digambarkan pada Gambar 3.7



Gambar 3.7 Diagram aktifitas manajemen data sales

5. Halaman sales

Sales yang telah melakukan proses login dapat melakukan proses input data report. Setelah data dimasukkan oleh sales, maka aplikasi akan mengecek apakah seluruh data telah terisi lengkap atau tidak. Jika data yang dimasukkan telah terisi lengkap, maka aplikasi akan menyimpan seluruh data tersebut pada basis data, hal ini menandakan bahwa proses masukkan data telah berhasil dilakukan dan pengguna kembali ke menu utama. Jika terdapat data yang dimasukkan tidak lengkap, maka pengguna akan mengulang proses memasukkan data hasil penilaian dari awal. Diagram aktifitas dari proses memasukkan data penjualan digambarkan pada Gambar 3.8.



Gambar 3.8 Diagram aktifitas memasukkan data penjualan

3.2.4.2 Kamus Data

Arus data yang terdapat pada Diagram Alur Data dapat dijelaskan menggunakan Kamus Data sebagai berikut:

1 *name: username*

where-used/how-used: masukan dan keluaran untuk proses baca *username password*.

masukan pada proses bandingkan *username password* dengan *Database*

description:

username = 20 string angka apa saja

2 *name: password*

where-used/how-used: masukan dan keluaran pada proses baca *username password*

masukan pada proses bandingkan *username password* dengan *database*

description:

password = 8 string angka apa saja

3 *name: valid*

where-used/how-used: keluaran untuk proses keabsahan *username password*

description:

valid = true

supplementary information:

tipe data: boolean

4 *name*: tidak sah

where-used/how-used: keluaran untuk proses validasi *username password*

description:

tidak valid = false

supplementary information:

tipe data: boolean

5 *name*: login admin

where-used/how-used: jalur aksi untuk menuju ke proses menu administrator

6 *name*: perintah

where-used/how-used: masukan bagi proses membaca perintah pemilih dan
masukan bagi proses membaca perintah administrator

7 *name*: tipe perintah

where-used/how-used: masukan bagi proses meminta pemrosesan perintah.

description:

tipe perintah = integer panjang 1 karakter

8 *name*: input

where-used/how-used: Jalur aksi untuk menuju ke proses baca data pengguna.

description:

hapus = integer panjang 1 karakter

9 *name*: edit

where-used/how-used: masukan untuk menuju ke proses baca *edit* data
pengguna.

description:

edit = integer panjang 1 karakter

10 *name*: hapus

where-used/how-used: masukan untuk menuju ke proses hapus data pengguna

description:

hapus = integer panjang 1 karakter

11 *name*: id

where-used/how-used: masukan bagi proses hapus data pengguna.

description:

id = integer dengan panjang 1 karakter

12 *name*: pilihan

where-used/how-used: masukan bagi proses membaca pilihan.

description:

pilihan = integer dengan panjang 1 karakter

13 *name*: tampilkan informasi

where-used/how-used: keluaran dari proses tampilkan pesan dan status.

description:

tampilkan informasi = [kata | gambar]

kata = text

14 *name*: *file login administrator*

where-used/how-used: masukan bagi proses bandingkan *username password*
dengan *file*

description:

file login administrator = [*username+password*]

username = 20 string huruf-angka

password = 8 string huruf-angka

3.2.5 Antarmuka Sistem

Penulis melakukan perancangan antarmuka aplikasi dengan berpedoman pada prinsip kemudahan pengguna (*user friendly*). Dengan penerapan konsep ini diharapkan pengguna mudah dalam memahami dan menggunakan aplikasi sehingga lebih mempermudah dalam melaksanakan pekerjaannya. Antarmuka sistem pada aplikasi ini dapat dijelaskan melalui struktur menu dan antarmuka.

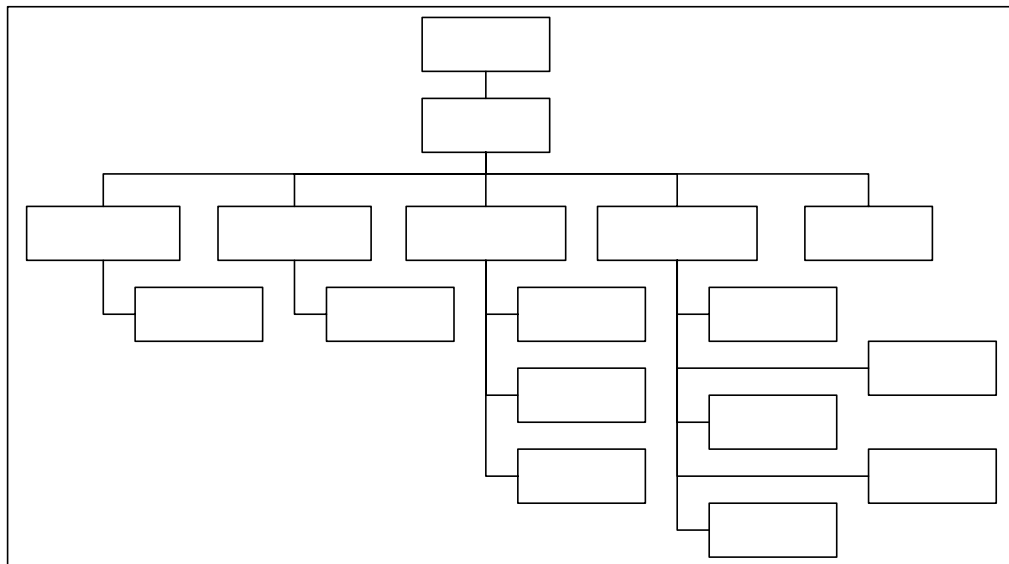
3.2.5.1 Struktur Menu

Aplikasi proteksi keamanan menggunakan user akses ini terdiri atas lima menu utama, yaitu:

1. About us, menu ini digunakan untuk menampilkan profil tentang perusahaan (PT. Lelco Trindo Graha Nuasantara)
2. Contact us, adalah menu untuk alamat-alamat dan nomor telepon yang dapat dihubungi untuk mendapatkan layanan dari perusahaan.
3. Buku tamu adalah menu untuk melampirkan kritik dan saran untuk perusahaan
4. Login admin adalah menu yang hanya dapat diakses oleh pengguna yang memiliki hak akses. Menu ini mempunyai lima halaman yang secara otomatis akan diakses oleh aplikasi sesuai level akses berdasarkan nama dan password yang di input pada saat login. Halaman pada login admin yaitu:
 - a. Halaman Super admin, adalah halaman yang hanya dapat diakses oleh super admin.
 - b. Halaman Top admin, adalah halaman yang hanya dapat diakses oleh top admin dan level di atasnya yaitu super admin .

- c. Halaman product admin, adalah halaman yang hanya dapat diakses oleh produk admin dan level di atasnya yaitu super admin dan top admin.
 - d. Halaman Sales admin, adalah halaman yang hanya dapat diakses oleh sales admin dan level di atasnya yaitu super admin dan top admin .
 - e. Halaman sales, adalah halaman yang diakses oleh sales untuk menginput laporan ke perusahaan.
5. Logout, yaitu menu untuk keluar atau mengakhiri penggunaan aplikasi.

Struktur menu aplikasi dapat digambarkan pada Gambar 3.9 berikut.



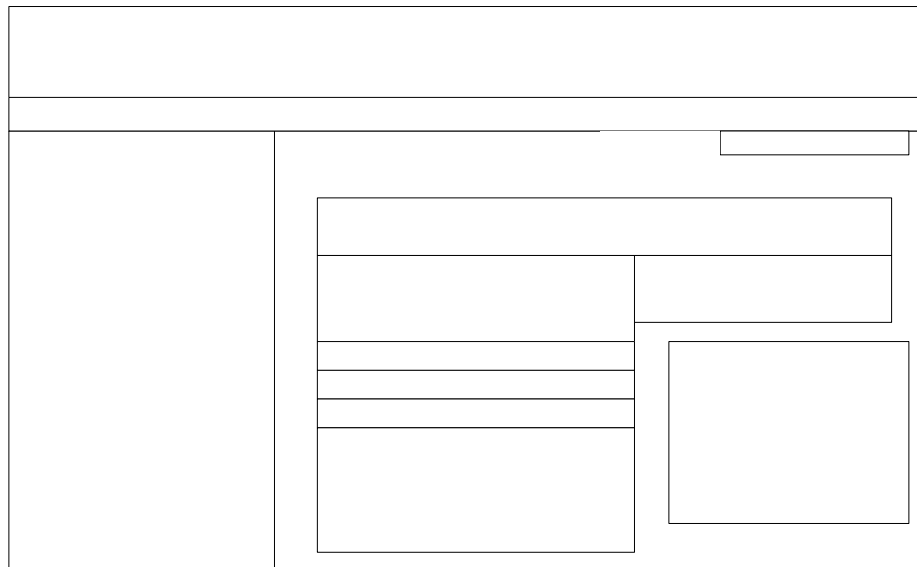
Gambar 3.9 Struktur menu aplikasi

3.2.5.2 Antarmuka Aplikasi

Antarmuka merupakan suatu bentuk komunikasi dan komponen penghubung antara pengguna dengan aplikasi. Perancangan antarmuka harus selalu berpedoman pada konsep *user friendly*. Rancangan antarmuka yang ada pada aplikasi adalah sebagai berikut:

1. Antarmuka Menu Utama.

Antarmuka menu utama adalah antarmuka awal dalam menggunakan aplikasi proteksi keamanan menggunakan user akses. Antarmuka ini berfungsi untuk tampilan semua pengguna yang masuk ke alamat situs perusahaan. Antarmuka ini terdiri atas empat menu yaitu menu home, produk, about us, contact us, buku tamu dan login admin. Selain itu, juga terdapat tabel-tabel yang disediakan untuk menyampaikan informasi perusahaan kepada pengguna, dalam hal ini seperti informasi kegiatan-kegiatan yang dilaksanakan perusahaan dan informasi karyawan yang berulang tahun. Rancangan antarmuka Menu utama dapat digambarkan pada Gambar 3.10.



Gambar 3.10 Antarmuka Menu Utama

2. Antarmuka produk

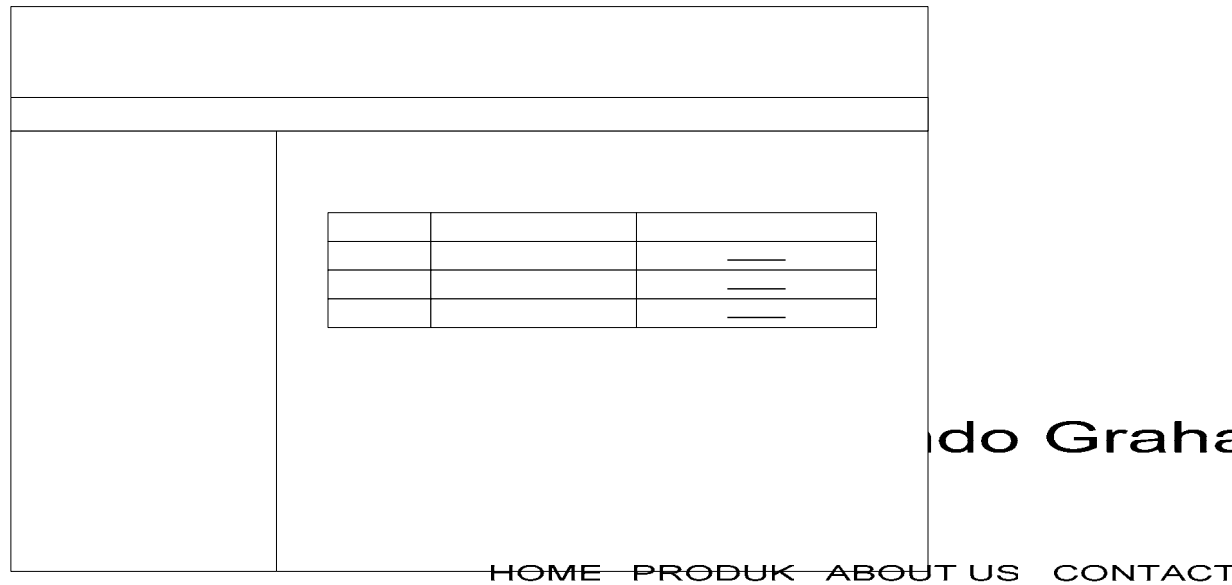
Setelah pengguna masuk ke menu utama dan memilih menu produk maka aplikasi akan menampilkan antarmuka produk. Antarmuka produk terdiri dari id produk, nama produk dan deskripsi produk.. Jika pengguna ingin mengakhiri

PT. Lelco Trindo Graha

HOME PRODUK ABOUT US CONTACT

PRODUK

penggunaan aplikasi, maka pengguna memilih menu home untuk menutup aplikasi. Rancangan antarmuka produk digambarkan pada Gambar 3.11.



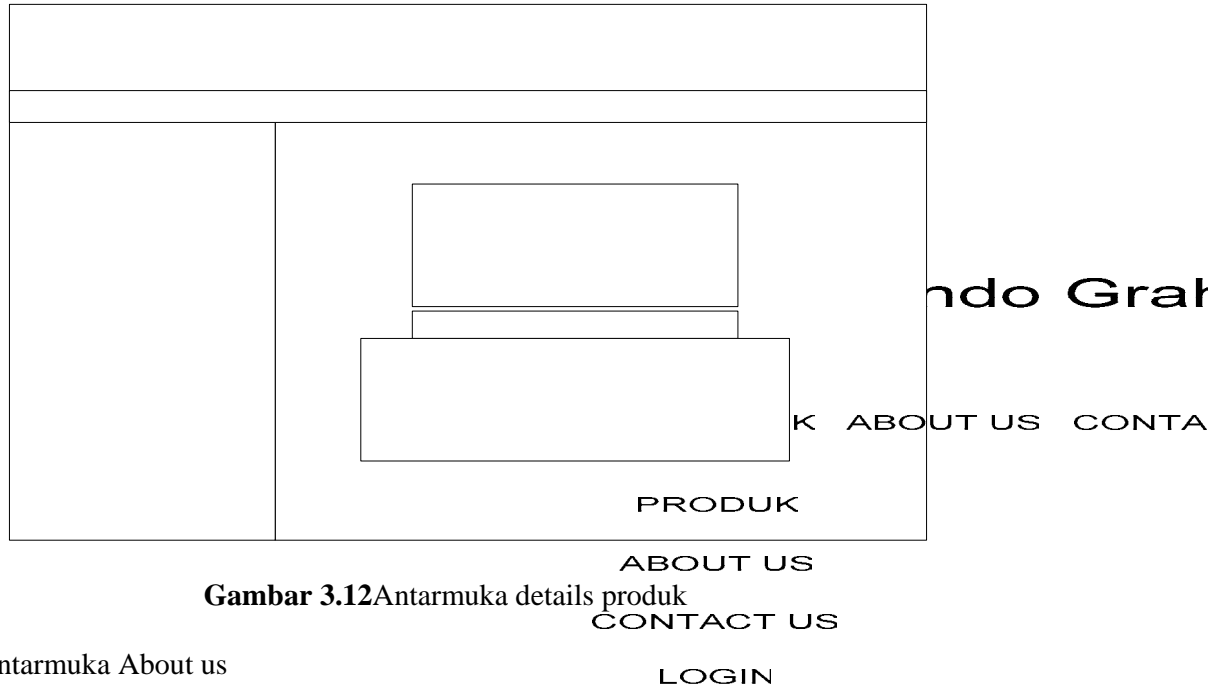
Gambar 3.11 Antarmuka menu produk

3. Antarmuka details produk.

Setelah pengguna masuk ke menu produk dan memilih menu details produk maka aplikasi akan menampilkan antarmuka details produk. Antarmuka details produk terdiri dari gambar produk, nama produk dan deskripsi produk.. Jika pengguna ingin mengakhiri penggunaan aplikasi, maka pengguna memilih menu home untuk menutup aplikasi. Rancangan antarmuka details produk digambarkan pada Gambar 3.12.

PRODUK
ABOUT US
CONTACT US
LOGIN

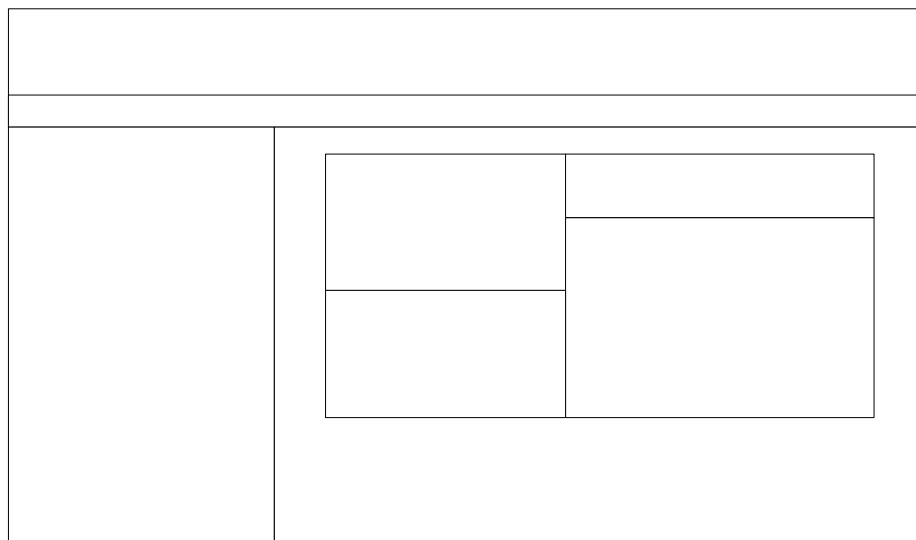
Id
xx
xx
xx



Gambar 3.12 Antarmuka details produk

4. Antarmuka About us

Pada antarmuka menu utama pengguna dapat memasuki menu about us yang berisi tentang company profile, merk-merk produk yang dijual perusahaan, dan perusahaan-perusahaan yang menjadi partner. Rancangan antarmuka about us digambarkan pada Gambar 3.13.



Gambar 3.13 Antarmuka about us

5. Antarmuka Buku Tamu.

Antarmuka buku tamu data adalah antarmuka untuk memasukkan data pengunjung yang ingin memberikan saran dan kritik untuk perusahaan. Data yang dimasukkan akan disimpan pada entitas buku tamu. Pengguna harus memasukkan identitas nama dan alamat e-mail sebelum memberikan komentar untuk dimasukkan ke basis data. Rancangan antarmuka buku tamu digambarkan pada Gambar 3.14.

Gambar 3.14 Antarmuka Buku Tamu

6. Antarmuka Contact.

Pada antarmuka menu utama pengguna dapat memasuki menu contact yang berisi tentang company profile, merk-merk produk yang dijual perusahaan, dan perusahaan-perusahaan yang menjadi partner. Rancangan antarmuka contact digambarkan pada Gambar 3.15.

PT. Lelco Trindo Graha

HOME PRODUK ABOUT US CONTACT

PRODUK

ABOUT US

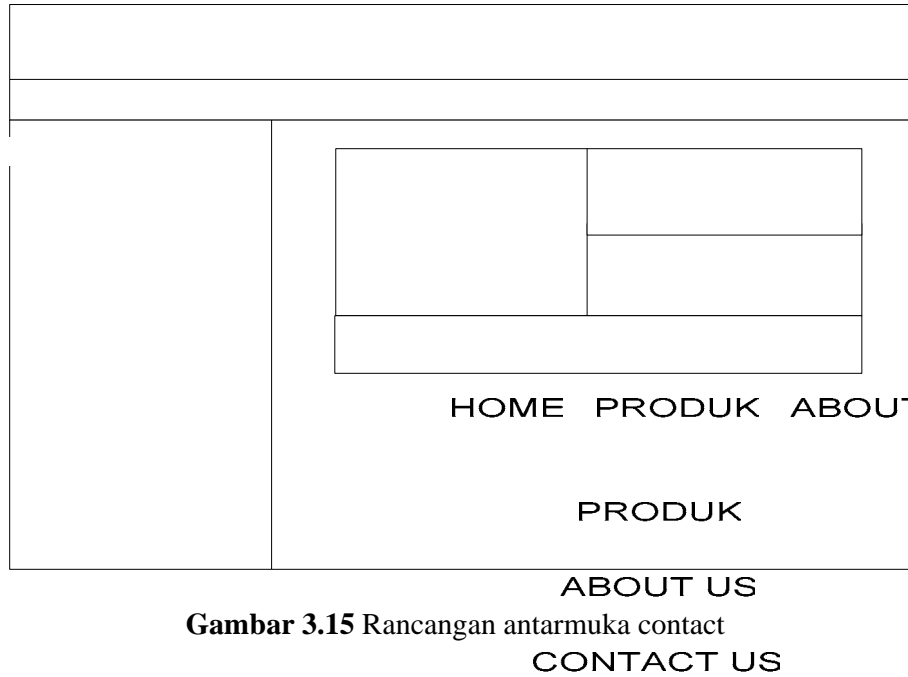
CONTACT US

LOGIN

Nama

E-mail

Komen



Gambar 3.15 Rancangan antarmuka contact

7. Antarmuka login.

LOGIN

Antarmuka login adalah antarmuka untuk administrator dan sales dalam perusahaan. Antarmuka ini berfungsi untuk membatasi akses penggunaan aplikasi, sehingga aplikasi hanya digunakan oleh pengguna yang berwenang, dalam hal ini administrator dan sales perusahaan. Antarmuka ini terdiri atas dua kotak masukan (*input box*) yaitu nama pengguna dan kata kunci. Selain itu, juga terdapat tombol login yang berfungsi untuk mengaktifkan prosedur program validasi nama pengguna dan kata kunci pada basis data dan tombol Batal untuk membatalkan proses login aplikasi dan kembali ke menu utama. Rancangan antarmuka login dapat digambarkan pada Gambar 3.16.

Gambar 3.16 Antarmuka login

Username :

8. Antarmuka update admin

Password :

Setelah pengguna berhasil melakukan proses login, maka akan masuk ke antarmuka menu utama administrator. Antarmuka menu utama administrator terdiri atas tujuh menu utama, yaitu menu update admin, menu update barang, menu lihat report, menu lihat produk, menu lihat buku tamu dan logout. Jika pengguna ingin mengakhiri penggunaan aplikasi, maka pengguna memilih menu Logout untuk menutup aplikasi. Rancangan antarmuka menu utama digambarkan pada Gambar 3.17.

login

Gambar 3.17 Rancangan antarmuka menu utama administrator

9. Antarmuka update admin

Antarmuka update admin menampilkan seluruh data administrator yang ada pada basis data. Antarmuka update admin adalah hak akses pada level super admin dan top admin. Antarmuka terdiri dari satu tabel dengan lima kolom yang berisi username, tipe user, password, hapus dan edit. Table digunakan untuk melihat data administrator dan sales yang ada di basis data. Dan pilihan hapus adalah untuk menghapus data administrator dan sales yang ada pada basis data. Pilihan edit adalah untuk mengubah data administrator dan sales yang ada pada basis data Rancangan antarmuka update admin digambarkan pada Gambar 3.18. Pada antar muka update admin juga terdapat sub menu new admin yang digunakan untuk menambah admin baru kedalam basis data dan digambarkan pada Gambar 3.18.

			—	—
			—	—
			—	—
			—	—

Gambar 3.18 Rancangan antarmuka update admin

The image shows a web interface for creating a new admin. On the left is a sidebar with several empty rectangular input fields. The main area contains a form with several input fields, a dropdown menu, and two buttons labeled 'Update admin' and 'Update barang'.

Gambar 3.19 Antarmuka new admin

10. Antarmuka update barang

Antarmuka update barang menampilkan **Lihat report** barang yang ada pada basis data perusahaan. Antarmuka update **Lihat produk** barang memiliki akses pada level super admin, top admin dan produk admin. **Lihat produk** Antarmuka update barang terdiri dari satu tabel dengan enam kolom yang berisi kode barang, nama barang, gambar, keterangan, hapus dan edit. Table digunakan untuk melihat data barang yang ada di basis data. Dan pilihan hapus adalah untuk menghapus data barang yang ada pada basis data. Pilihan edit adalah untuk **Logout** mengubah data barang yang ada pada basis data. Rancangan antarmuka update barang digambarkan pada Gambar 3.20. Pada antar muka update barang juga terdapat sub menu tambah barang yang digunakan untuk menambah barang baru kedalam basis data. Antarmuka update barang digambarkan pada Gambar 3.20.

The interface for updating goods consists of a sidebar on the left with several input fields and a main content area. The main area contains a table with 6 columns and 4 rows. Below the table are two buttons: 'Update admin' and 'Update barang'.

Gambar 3.20 Rancangan antarmuka update barang

The interface for adding goods consists of a sidebar on the left with several input fields and a main content area. The main area contains several input fields, a text area with a scroll bar, and two buttons at the bottom.

Gambar 3.21 Antarmuka tambah barang

11. Antarmuka Lihat laporan

Antarmuka Lihat laporan menampilkan seluruh data laporan yang ada pada basis data perusahaan. Antarmuka lihat laporan adalah hak akses pada level super admin, top admin dan sales admin. Antarmuka lihat laporan terdiri dari satu

PT. Lelco Trindo Graha

Update admin

Use

tabel dengan empat kolom yang berisi kode sales, nama sales, data sales, dan cek data. Table digunakan untuk melihat data laporan yang telah masuk kedalam basis data. Pada antarmuka lihat laporan juga terdapat sub menu compile data yang digunakan untuk membuat rangkuman laporan penjualan berdasarkan data laporan sales yang telah masuk ke basis data perusahaan. Antarmuka lihat laporan digambarkan pada Gambar 3.22.

The image shows a wireframe of a report viewing interface. It features a header area at the top. Below the header, there is a vertical sidebar on the left containing several rectangular input fields. The main content area on the right contains a table with four columns and four rows. Below the table, there is a single rectangular button.

Gambar 3.22 Rancangan antarmuka lihat laporan

12. Antarmuka sales

Antarmuka sales akan ditampilkan otomatis setelah aplikasi mengenali pengguna sebagai sales perusahaan berdasarkan proses login. Pada antarmuka sales dibatasi pada akses menu tentang kami, buku tamu, contact dan input laporan. Sales akan memasukkan laporan penjualan melalui menu input laporan. Rancangan antarmuka sales dapat digambarkan pada Gambar 3.23.

PT. Lelco Trindo Graha

Update admin

User

Update barang

Kode sales

xxx

Lihat report

xxx

xxx

Lihat produk

xxx

Lihat buku tamu

The wireframe shows a rectangular layout. On the left side, there is a vertical sidebar containing five rectangular input fields. The main content area on the right is mostly empty, with the text "Tentang kami" centered at the bottom.

Gambar 3.23 Rancangan antarmuka sales
Buku tamu

13. Antarmuka Input Laporan

Contact us

Antarmuka input laporan adalah antarmuka yang berfungsi untuk memasukkan data laporan ke dalam basis data. Antarmuka ini terdiri atas 3 jenis Input laporan masukkan data yaitu nama sales, kode sales dan produk. Pada masukkan produk, sales hanya bisa memilih produk dari basis data yang telah dimasukkan oleh admin produk. Rancangan antarmuka hapus seluruh data digambarkan pada Gambar 3.24.

The wireframe shows a rectangular layout. On the left side, there is a vertical sidebar containing five rectangular input fields. The main content area is divided into three sections: a central area with three green checkmarks, a right sidebar with three rectangular input fields, and two rectangular input fields at the bottom center.

Gambar 3.24 Rancangan antarmuka input laporan

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Setelah tahap analisis dan perancangan pada bab III, maka tahap selanjutnya yaitu tahap implementasi. Pada bab ini akan dibahas mengenai implementasi tentang metode keamanan data menggunakan metode MD-5 (*Message Digest Algorithm 5*), pembuatan basis data dan implementasi perancangan layar.

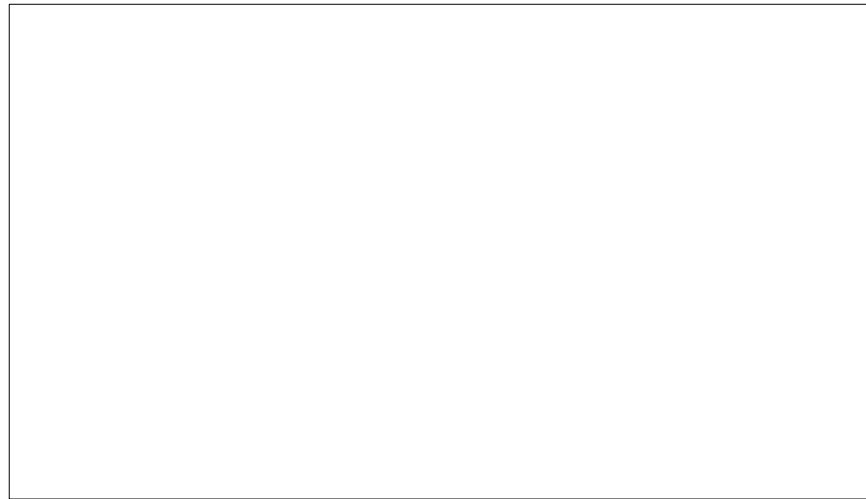
4.1.1 Implementasi Metode MD-5 (*Message Digest Algorithm 5*).

Pada implementasi metode MD-5 (*Message Digest Algorithm 5*) ini perangkat lunak yang digunakan yaitu phptriad. Pada aplikasi ini metode MD-5 hanya digunakan untuk menjaga keamanan dari password user dan administrator. metode MD-5 (*Message Digest Algorithm 5*) di masukkan pada halaman Top-admin hal ini dilakukan sebagai identifikasi dan pemberian akses dalam melihat password user tergantung pada level user yang menggunakan aplikasi.

4.1.2 Implementasi pemakaian Metode MD-5 pada aplikasi

Tb_admin terdiri dari tiga kolom data, kolom pertama berisi data *username*, kolom kedua berisi data *password*, dan kolom ketiga berisi data nama admin.

Kode 4.1 merupakan implementasi basis data tb_admin.



Kode 4.1 Implementasi pemakaian Metode MD-5 pada aplikasi.

Kode program dari nomor 2 sampai nomor 7 akan menjelaskan: Sistem akan melakukan koneksi ke halaman KONEKSI.php dan server mysql akan mengenali tipe user. Kode program nomor 8 sampai no 16 akan menjelaskan: Jika data uname, dan tipe user telah dikenali oleh sistem sebagai super admin dan top admin maka password dari pengguna akan terlihat tanpa di enkrip dengan MD-5 tetapi apabila pengguna bukan master admin dan top admin maka pengguna tersebut tidak akan bisa melihat password dari pengguna yang lain karena karakter password telah diubah menggunakan metode MD-5.

```

1. <?
2. include "KONEKSI.php"
3. //mysql_connect("localhost")
4. //mysql_select_db("indri")
5. $cari="SELECT * FROM"
6. $hasil=mysql_query($cari)
7. while($data=mysql_fetch_row($hasil))
8. {
9.     echo "<tr>";
10.    echo("<td align=center>";
11.    echo("<td align=center>";
12.    $pass=md5($pswd);
13.    echo("<td align=center>";
14.    echo("</tr>");
15.    }
16.    ?>

```


4.2 Implementasi Pembuatan Basis Data

Pada implementasi pembuatan basis data ini perangkat lunak yang digunakan yaitu phptriad. Pada implementasi ini hanya menampilkan basis data dari tabel-tabel yang fungsional saja, seperti: `tb_administrator`, `tb_bukutamu`, `tb_produk`, `tb_report` dan `tb_tipeuser`. Masing-masing tabel tersebut telah dijelaskan pada tahap perancangan pada bab III.

4.2.1 Implementasi Pembuatan Basis Data `Tb_administrator`

`Tb_administrator` terdiri dari empat kolom data, kolom pertama berisi data *userid* dengan lebar maksimum 5 karakter, kolom kedua berisi data *user* dengan lebar maksimum 20 karakter, dan kolom ketiga berisi data *password* dengan lebar maksimum 8 karakter dan keempat berisi data *tipeuser* dengan lebar maksimum 20 karakter. Gambar 4.1 merupakan implementasi basis data `tb_administrator`.

Database `indrisql` - table `administrator` running on localhost

Showing records 0 - 5 (5 total)

SQL-query: [Edit]
SELECT * FROM 'administrator' LIMIT 0, 30

Show: 30 rows starting from 0
in horizontal mode and repeat headers after 100 cells

	userid	user	password	tipeuser
Edit Delete	555	leo	555	Sales
Edit Delete	333	kan2	888	Product Admin
Edit Delete	444	otong	444	Sales Admin
Edit Delete	111	edwin	111	Top Admin
Edit Delete	222	topoy	999	Top Admin

Show: 30 rows starting from 0
in horizontal mode and repeat headers after 100 cells

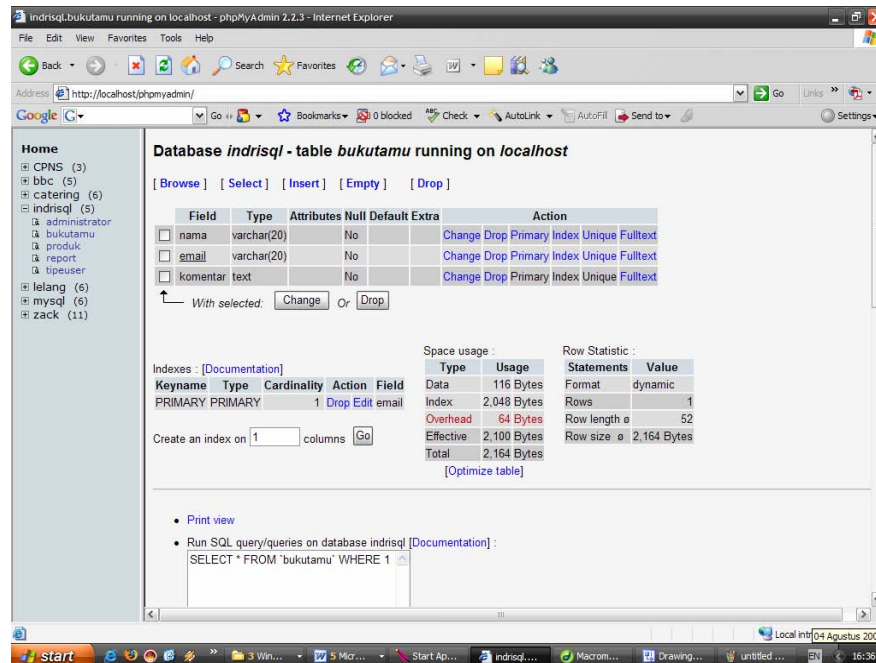
[Insert new row](#)

Gambar 4.1 Implementasi Pembuatan Basis Data Tabel Adminisrtator

4.2.2 Implementasi Pembuatan Basis Data Tb_Buku tamu

Tb_bukutamu terdiri dari tiga kolom data, kolom pertama berisi nama dengan lebar maksimum 20 karakter, kolom kedua berisi email dengan lebar maksimum 20 karakter, dan kolom ketiga berisi komentar dengan menggunakan text.

Gambar 4.2 merupakan implementasi basis data tb_bukutamu.

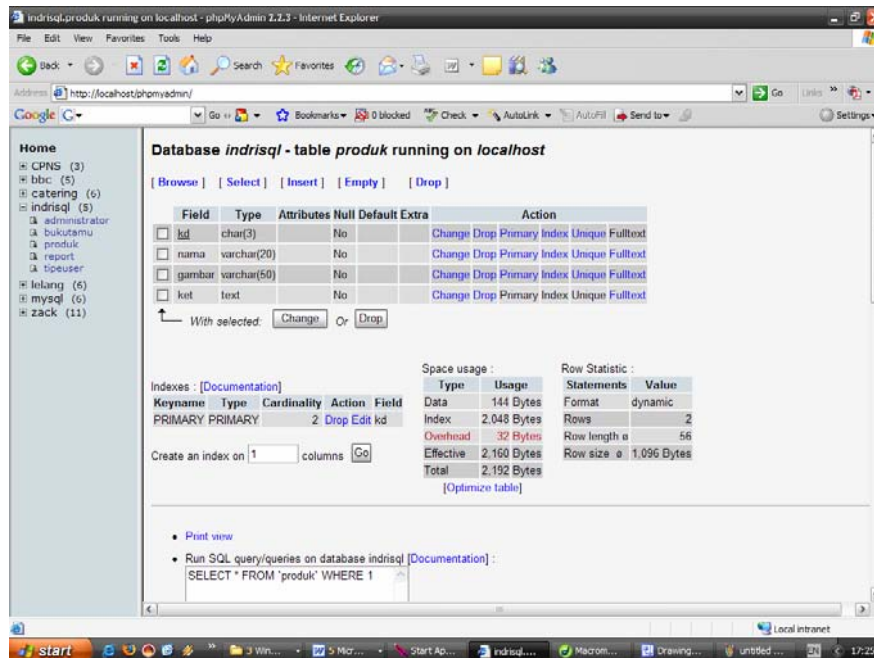


Gambar 4.2 Implementasi Pembuatan Basis Data Tabel Buku tamu.

4.2.3 Implementasi Pembuatan Basis Data Tb_Produk

Tb_produk terdiri dari empat kolom data, kolom pertama berisi data kd_produk dengan lebar maksimum 3 karakter, kolom kedua berisi data nama dengan lebar maksimum 20 karakter, kolom ketiga berisi data gambar dengan lebar maksimum 50 karakter, kolom keempat berisi data keterangan dengan menggunakan text. Tipe text digunakan untuk memberikan keterangan tentang produk yang di tampilkan.

Gambar 4.3 merupakan implementasi basis data tb_produk.

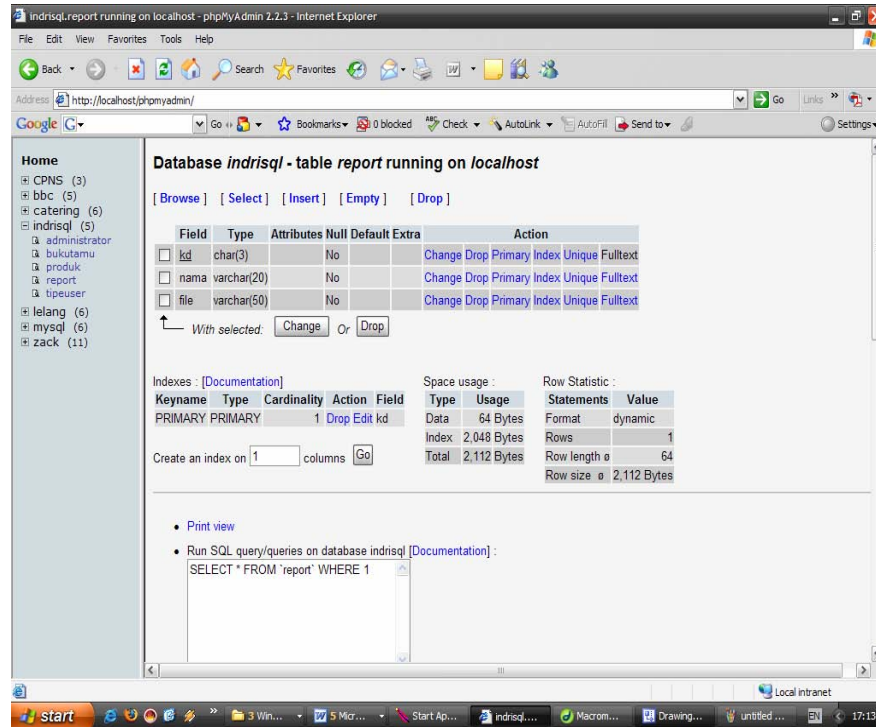


Gambar 4.3 Implementasi Pembuatan Basis Data Tabel Produk.

4.2.4 Implementasi Pembuatan Basis Data Tb_report

Tb_report terdiri dari tiga kolom data, kolom pertama berisi data kd_report dengan lebar maksimum 3 karakter, kolom kedua berisi data nama dengan lebar maksimum 20 karakter, dan kolom ketiga berisi data file dengan lebar maksimum 50 karakter.

Gambar 4.4 merupakan implementasi basis data tb_report.

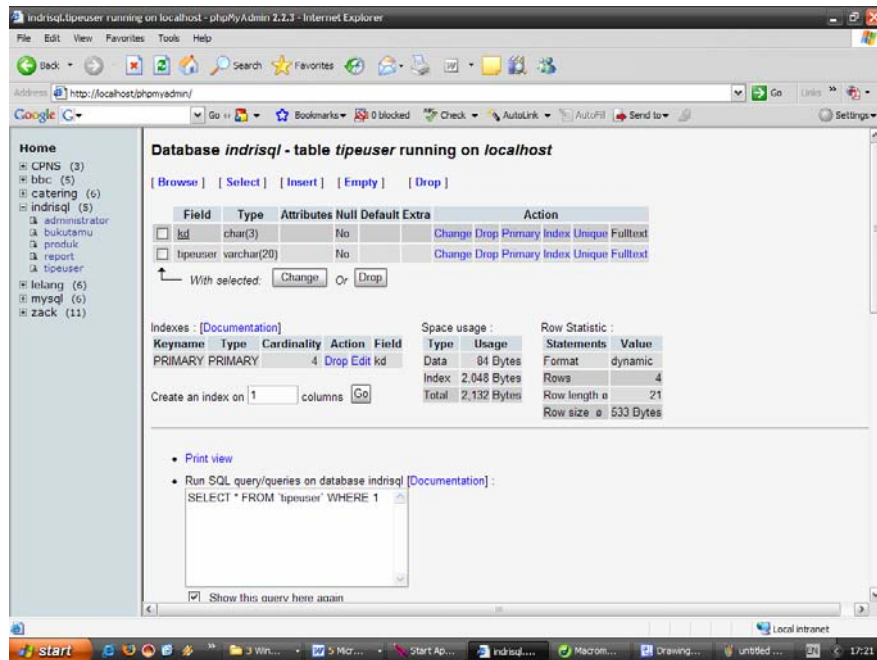


Gambar 4.4 Implementasi Pembuatan Basis Data Tabel report

4.2.5 Implementasi Pembuatan Basis Data Tb_tipeuser

Tb_tipeuser terdiri dari dua kolom data, kolom pertama berisi kd_tipeuser dengan lebar maksimum 3 karakter, kolom kedua berisi tipeuser dengan lebar maksimum 20 karakter.

Gambar 4.5 merupakan implementasi basis data tb_tipeuser.



Gambar 4.5 Implementasi Pembuatan Basis Data Tabel tipeuser

4.3 Implementasi Perancangan Layar

Pada implementasi perancangan layar akan menampilkan rancangan layar dan potongan program, seperti: menu utama, menu login, menu admin, menu tambah admin, menu produk, menu tambah produk, dan menu buku tamu. Masing-masing perancangan layar tersebut telah dijelaskan pada bab III.

4.3.1 Implementasi Perancangan Layar Menu Utama

Implementasi perancangan layar menu utama akan menampilkan halaman utama yang berisi berita-berita atau pengumuman dari perusahaan.

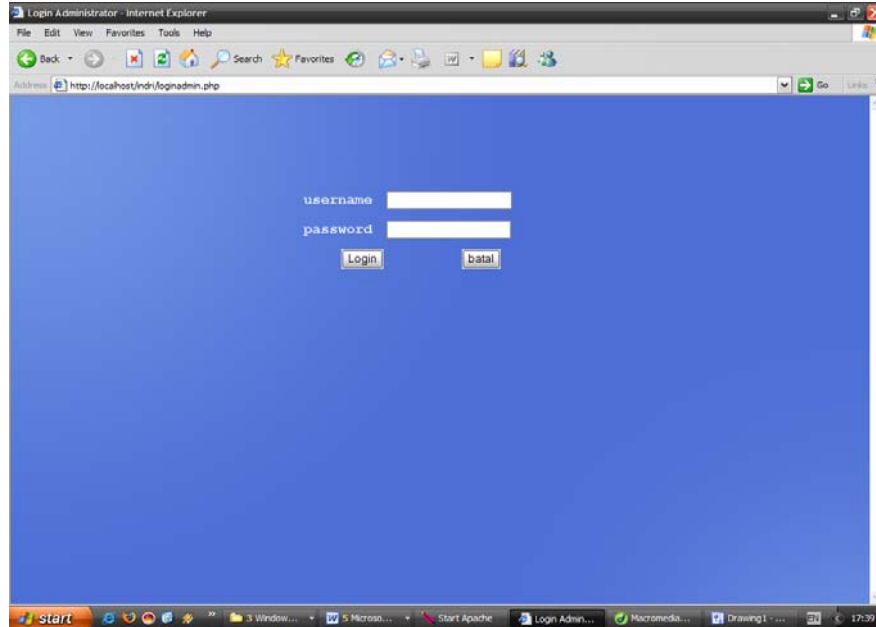
Gambar 4.6 merupakan implementasi perancangan layar menu utama.



Gambar 4.6 Implementasi Perancangan Menu Utama.

4.3.2 Implementasi Perancangan Layar Login Administrator

Implementasi perancangan layar login peserta dibuat dengan tujuan untuk membatasi siapa saja yang berhak mengakses aplikasi ini, sistem akan memberi ijin kepada siapapun untuk mengakses menu administrator ini jika sudah melakukan proses login terlebih dahulu. Gambar 4.7 merupakan implementasi perancangan layar login administrator.



Gambar 4.7 Implementasi Perancangan Layar Login Administrator.

Kode program 4.2. akan menggambarkan proses yang terjadi setelah administrator melakukan login.

Kode 4.2. Proses login administrator

```

1.      <?
2.      if($Login)
3.      {
4.          if($Saman =='indri' && $Login =='123')
5.          {
6.              $Saman="indri";
7.              $Stipe="Super Admin";
8.              session_start();
9.              session_register("nama","tipe");
10.         echo "<script>document.location=\"menu_update_Data_new.php\"</script>";
11.         exit();
12.         }
13.         else
14.         {
15.             $nama=$Saman;
16.             $psw=$Login;
17.             include "KONEKSI.php";
18.             $perintah="select * from Administrator where user= '$nama' and password = '$psw'";
19.             $hasil=mysql_query($perintah) or die(mysql_error());
20.             $data=mysql_fetch_array($hasil);
21.             if ($data)
22.             {
23.                 $nama="$data[user]";
24.                 $tipe="$data[tipeuser]";
25.                 session_start();
26.                 session_register('nama','tipe');
27.                 echo "<script>document.location=\"Menu_Update_Data_new.php\"</script>";
28.                 exit();
29.             }e
30.             else
31.             {
32.                 echo "<script>alert(\"Anda Tidak Berhak..!\");history.go(-1)</script>";
33.                 exit();
34.             }
35.             ?>
36.         </script>
37.         <form action="cek1.php" method="post" enctype="multipart/form-data" >
38.         <strong>username</strong></font></div> </td>
39.         <input type="text" name="aman">
40.         <input name="login" type="password" id="pswd">
41.         <input type="submit" name="Login" value="Login">
42.         <input type='button' onclick=location='index.php' value="batal">
43.         <table width="787" border="0" cellpadding="0" cellspacing="0">
44.         <!--DWLayoutDefaultTable-->
45.         <tr>
46.         </html>

```

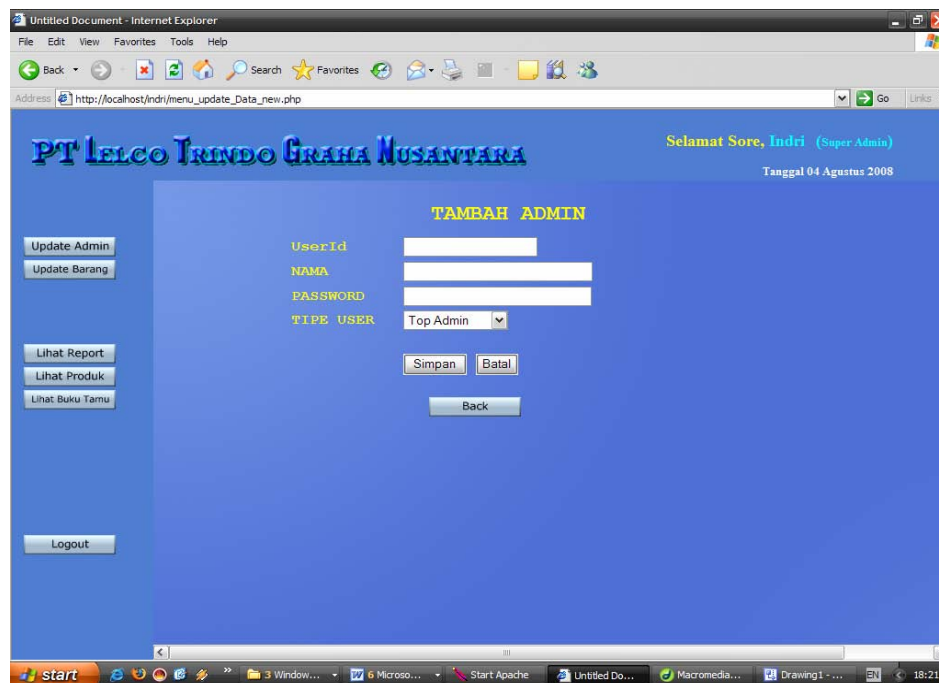
Kode program nomor 2 adalah proses yang dilakukan oleh sistem dalam hal ini sistem akan melaksanakan proses login administrator.

Kode program dari nomor 5 sampai nomor 12 akan menjelaskan proses login super administrator yaitu apabila pengguna memasukkan *username* “indri” dan memasukkan *password* “123” maka program akan secara otomatis akan mengenali

pengguna sebagai super admin. Kode program 13 sampai dengan 29 akan menjelaskan: apabila pengguna memasukkan *username* dan *password* selain dari nama yang diatas maka aplikasi akan melakukan koneksi ke *server* basis data *mysql* untuk menentukan nama dan *tipeuser* berdasarkan data yang ada pada basis data *mysql*. Pada kode program 30 sampai dengan 34 menjelaskan jika koneksi tidak berhasil, maka sistem akan menampilkan pesan kesalahan. Setelah koneksi berjalan baik, sistem akan mengaktifkan basis data, dalam aplikasi ini basis data yang digunakan adalah *db_indrisql*.

4.3.3 Implementasi Perancangan Layar Menu Tambah Admin

Implementasi perancangan menu admin digunakan oleh super administrator dan top administrator untuk menambah data administrator baru pada basis data perusahaan. Gambar 4.8 merupakan Implementasi perancangan menu tambah admin.



Gambar 4.8 Implementasi Perancangan menu tambah admin

Kode program 4.3. merupakan gambaran yang terjadi pada saat super administrator dan top administrator ingin menambah administrator baru pada perusahaan.

Kode 4.3. Proses tambah administrator

```

1. <?
2. session_start();
3. if(!(session_is_registered("tipe"))
4. {
5. echo "<script>alert('\Level Anda Bukan Top Admin\");history.go(-1)</script>";
6. exit;
7. }
8. else
9. { if ($tipe!="Super Admin" && $tipe!="Top Admin")
10. {echo "<script>document.location=\"edit_admin.php\"</script>";exit;}}
11. ?>
12. <html>
13. <head>
14. <title>Tambah Admin Baru</title>
15. </script>
16. <form name="form1" method="post" action="NewAdmin.php"
17. </table>
18. <input type="text" name="userid" maxlength="10">
19. <input type="text" name="user" size="30" maxlength="10">
20. <input type="pwd" name="password" size="30" maxlength="10">
21. <td colspan="2"><select name="tipeuser" id="tipeuser">
22. <option value="Top Admin">Top Admin</option>
23. <option value="Product Admin">Product Admin</option>
24. <option value="Sales Admin">Sales Admin</option>
25. <option value="Sales">Sales</option>
26. </select>
27. <input name="tambah" type="submit"
28. value="Simpan">
29. <input name="reset" type="reset" value="Batal">

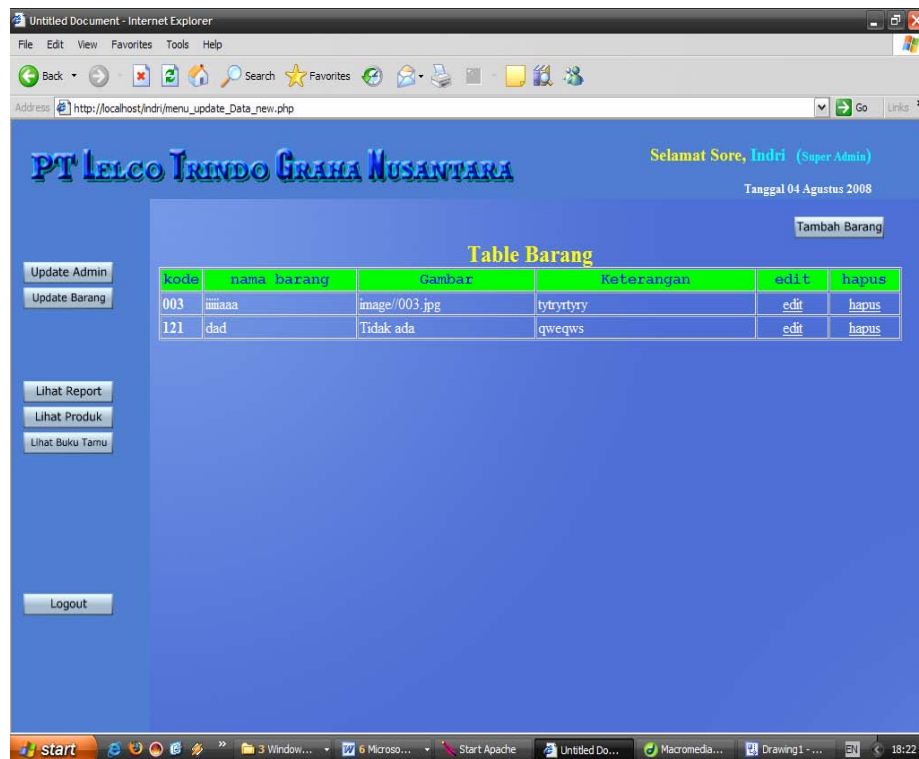
```

Kode program nomor 1 sampai dengan 7 menjelaskan apabila pengguna bukan bertipe super admin dan top admin maka program akan menampilkan pesan peringatan dan akan kembali ke menu utama. Kode program nomor 8 sampai dengan 10 menjelaskan bahwa aplikasi akan menampilkan halaman menu admin setelah pengguna telah dikenali sebagai super admin dan top admin. Kode program nomor 11 sampai dengan nomor 26 adalah tampilan input data administrator baru yang akan ditambahkan dalam basis data perusahaan yaitu iduser, nama, password

dan tipeuser. Kode program nomor 27 sampai nomor 29 adalah proses tombol submit yaitu memasukkan data kedalam basis data dan batal untuk membatalkan proses penambahan administrator baru.

4.3.4 Implementasi Perancangan Layar Edit Barang

Implementasi perancangan layar edit barang digunakan untuk merubah dan menghapus data produk yang ada dalam perusahaan. Hak akses untuk menu edit barang diberikan pada admin produk dan level-level di atasnya yaitu super administrator dan top administrator. Gambar 4.9 merupakan Implementasi perancangan layar edit barang.



Gambar 4.9 Implementasi Perancangan Layar edit barang.

Kode program 4.4. merupakan potongan program perancangan layar edit barang.

Kode 4.4. Proses edit barang

```

1.      <?
2.      session_start();
3.      if(!(session_is_registered("tipe"))
4.      {echo "<script>alert(\"Anda Tidak punya hak untuk mengakses halaman ini\
\");history.go(-1)</script>";exit;}
5.      ?>
6.      <html><head><title>Update Isi Tabel barang</title></head>
7.      <body background="newmark1.jpg">
8.      <param name="movie" value="tambah barang.swf">
9.      include "KONEKSI.php";
10.     $cari="SELECT * FROM produk order by right(kd,length(kd)-1)";
11.     $hasil=mysql_query($cari);
12.     while($data=mysql_fetch_array($hasil))
13.     echo "<tr>";
14.     echo("<td align=left><font color=#ffffff><b>$data[kd]</b></font></td>");
15.     echo("<td align=left><font color=#ffffff>$data[nama]</font></td>");
16.     echo("<td align=left><font color=#ffffff>$data[gambar]</font></td>");
17.     echo("<td align=left><font color=#ffffff>$data[ket]</font></td>");
18.     echo("<td align=center><a href=edit_barang.php?kode=$data[kd]><font
19.     echo("<td align=center ><a href=hapus_barang.php?kode=$data[kd] title='hati-
hati'><font color=#ffffff>hapus</font></a></td>");
20.     echo("</tr>");
21.     </table>
22.     <p><b><a href="Menu_Update_Data_new.php"></a></b></p>
23.     </center></body></html>

```

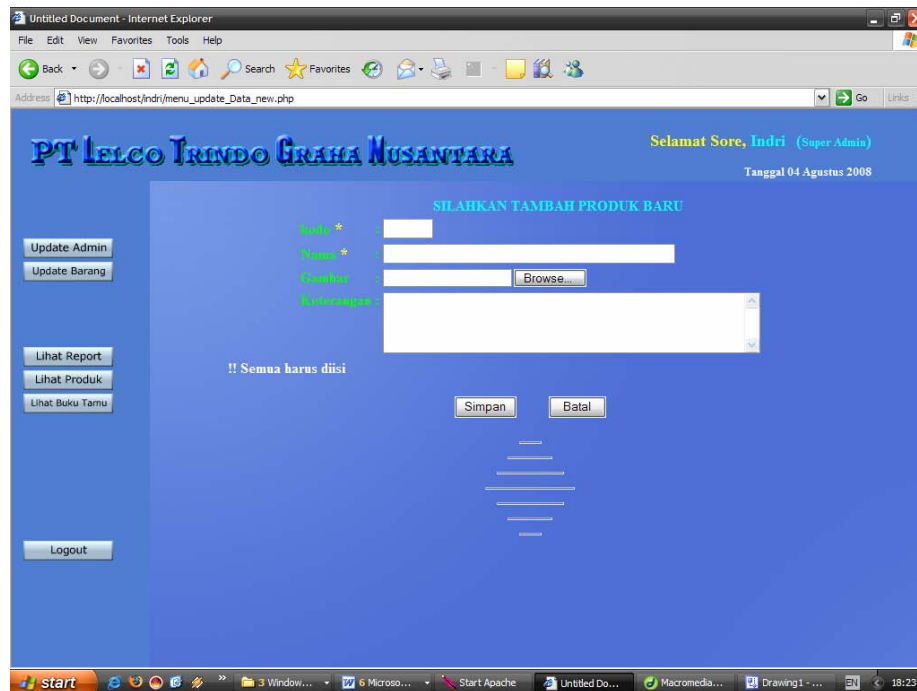
Kode program nomor 1 sampai dengan 5 menjelaskan: Sistem akan melakukan pemeriksaan ke basis data untuk mencari data pengguna yang memasuki menu ini. Hal ini di lakukan untuk menentukan hak akses pengguna yang sedang menggunakan aplikasi tersebut.

Kode program nomor 6 sampai nomor 20 menjelaskan perancangan dari layar edit barang. Pada layar edit barang ditampilkan data barang, gambar barang dan keterangan yang berkaitan dengan barang tersebut administrator lalu dapat mengubah atau menghapus data barang dengan memilih submenu edit atau hapus yang terdapat

pada tabel. Sedangkan kode program nomor 21 sampai nomor 23 menjelaskan bahwa sistem akan kembali ke menu administrator setelah proses edit barang selesai.

4.3.5 Implementasi Perancangan Layar Menu Tambah Produk

Implementasi perancangan layar menu tambah produk digunakan oleh administrator yang memiliki hak akses untuk menambah produk baru kedalam basis data perusahaan, admin akan masuk ke halaman menu tambah produk dengan memilih sub menu tambah barang pada layar edit barang.. Gambar 4.10 merupakan Implementasi perancangan tambah produk.



Gambar 4.10 Implementasi Perancangan Layar menu tambah produk.

Kode program 4.5. merupakan potongan program implementasi perancangan layar tambah produk.

Kode 4.5. Proses halaman tambah produk

```

1.      <?
2.      session_start();
3.      if(!session_is_registered("tipe")
4.      {echo "<script>alert(\"Anda Tidak punya hak untuk mengakses halaman
      ini\");history.go(-1)</script>");}
5.      ?>
6.      <title>Tambah Isi Tabel Barang </title></head>
7.      <body background="newmark1.jpg">
8.      <form name="form" method="post" action="newbarang.php">
9.      <b><font color="#00FFFF">SILAHKAN TAMBAH PRODUKBARU</font></b>
10.     <td width="26%"><font color="#00FF00"><b>kode</b><font color="#FFFF00" size="4">
11.     <input type="text" name="nama" size="50" maxlength="50"> </td>
12.     <font color="#00FF00">Gambar </font></b> </td>
13.     <input align="top" type="file" name="gambar"> </td>
14.     <b><font color="#00FF00">Keterangan</font></b> </td>
15.     Semua </font> <font color="#FFFFFF">harus diisi</font>
16.     </strong></font></td>
17.     <input type="submit" name="tambah" value="Simpan" onClick="return validate ()">
18.     <input type="reset" name="ulang" value=" Batal " >
19.     </body></html>

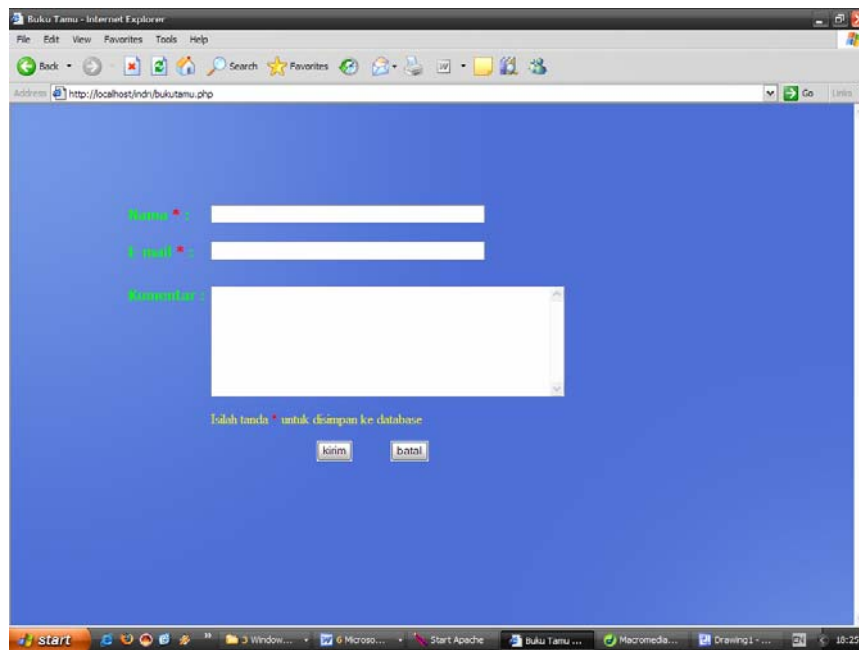
```

Kode program nomor 1 sampai dengan 5 menjelaskan: Sistem akan melakukan pemeriksaan ke basis data untuk mencari data pengguna yang memasuki menu ini. Hal ini dilakukan untuk menentukan hak akses pengguna yang sedang menggunakan aplikasi tersebut.

Kode program nomor 6 sampai nomor 16 menjelaskan perancangan dari layar tambah produk. Pada layar tambah produk terdapat field-field yang harus diisi oleh administrator, kode barang adalah nomor kode produk baru yang harus dimasukkan sebagai identitas produk pada basis data, nama barang adalah nama produk baru perusahaan, di halaman ini administrator juga memasukkan gambar produk untuk keterangan visual produk dan keterangan yang berkaitan dengan barang tersebut. Kode program nomor 17 sampai nomor 19 adalah proses tombol simpan yaitu memasukkan data kedalam basis data dan batal untuk membatalkan proses penambahan administrator baru.

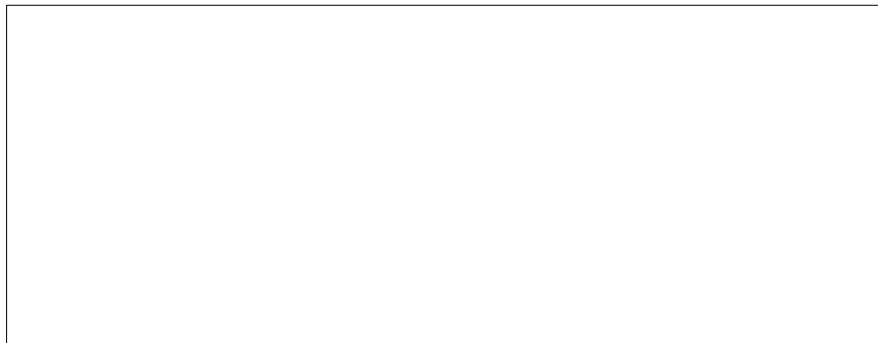
4.3.6 Implementasi Perancangan Layar Menu Buku Tamu

Implementasi perancangan layar menu buku tamu adalah sebagai sarana kepada pengunjung situs perusahaan untuk memberikan komentar atau kritik kepada perusahaan atau produk yang di jual oleh perusahaan. Gambar 4.11 merupakan Implementasi perancangan layar menu buku tamu.



Gambar 4.11 Implementasi perancangan layar menu buku tamu.

Kode 4.6. Proses halaman menu buku tamu



Kode program nomor 1 sampai dengan 5 menjelaskan: field yang diisi oleh pengunjung berupa nama, e-mail pengunjung dan komentar yang disampaikan.

Kode program nomor 6 menjelaskan field yang harus diisi sebelum pengunjung dapat mengirimkan komentar, apabila field yang menggunakan tanda * tidak diisi maka program akan memunculkan tanda peringatan yang meminta pengunjung untuk mengisinya.. Kode program nomor 7 sampai nomor 9 adalah proses tombol kirim yaitu memasukkan data kedalam basis data buku tamu dan batal untuk membatalkan proses pengisian buku tamu.

4.4 Pengujian

Setelah tahap implementasi, tahap selanjutnya adalah tahap pengujian. Pada tahap ini akan dibahas mengenai lingkungan pengujian, skenario pengujian, dokumen hasil pengujian dan analisis hasil pengujian.

4.4.1 Lingkungan Pengujian

Lingkungan pengujian pada aplikasi ini meliputi perangkat keras dan perangkat lunak, spesifikasi masing-masing lingkungan tersebut adalah sebagai berikut:

1. Spesifikasi Perangkat Lunak, meliputi:
 - a. Sistem operasi yang digunakan pada aplikasi ini adalah Microsoft Windows XP.
 - b. Macromedia Dreamweaver MX 2004 digunakan untuk membuat program aplikasi ini.
 - c. Phptriad digunakan untuk menjalankan aplikasi ini

- d. Phpmysql digunakan untuk membuat basis data.
 - e. Apache digunakan sebagai localhost.
 - f. Microsoft Internet Explorer 4.0 digunakan sebagai *browser* internet.
2. Spesifikasi Perangkat Keras, meliputi
- a. Komputer dengan prosesor intel Pentium 1.8 GHz.
 - b. Memori 256 MB.
 - c. VGA 64 MB.
 - d. Monitor 15'
 - e. Keyboard dan Mouse

4.4.2 Skenario Pengujian

Skenario pengujian meliputi pengujian perangkat lunak terhadap fungsionalitas aplikasi yang dibangun berfungsi untuk melakukan pengujian proses yang terjadi pada perangkat lunak yang dibangun dengan proses yang terjadi didalam sistem dan menguji kelayakan aplikasi dalam sebuah situs yang akan digunakan.

Metode yang digunakan dalam aplikasi ini adalah *black-box testing*. Metode ini sesuai dengan aplikasi yang akan dibangun, karena metode *black-box testing* melakukan pengujian dengan cara memberikan sejumlah masukan pada program aplikasi yang akan diproses sesuai dengan kebutuhan fungsionalnya untuk menghasilkan keluaran yang diinginkan.

Apabila keluaran proses tidak menghasilkan sesuai yang diinginkan, maka program aplikasi masih terdapat kesalahan-kesalahan yang harus diperbaiki pada program aplikasi tersebut. Apabila keluaran proses menghasilkan kebutuhan fungsionalnya, maka program aplikasi tersebut benar.

Web browser Internet Explorer digunakan untuk melakukan pengujian pada program aplikasi, dengan cara mengetikkan alamat URL sebagai berikut: <http://localhost/indri/index.php>. *Black-box testing* melakukan pengujian pada program aplikasi dimulai dari menu utama sampai semua menu yang terdapat pada program aplikasi.

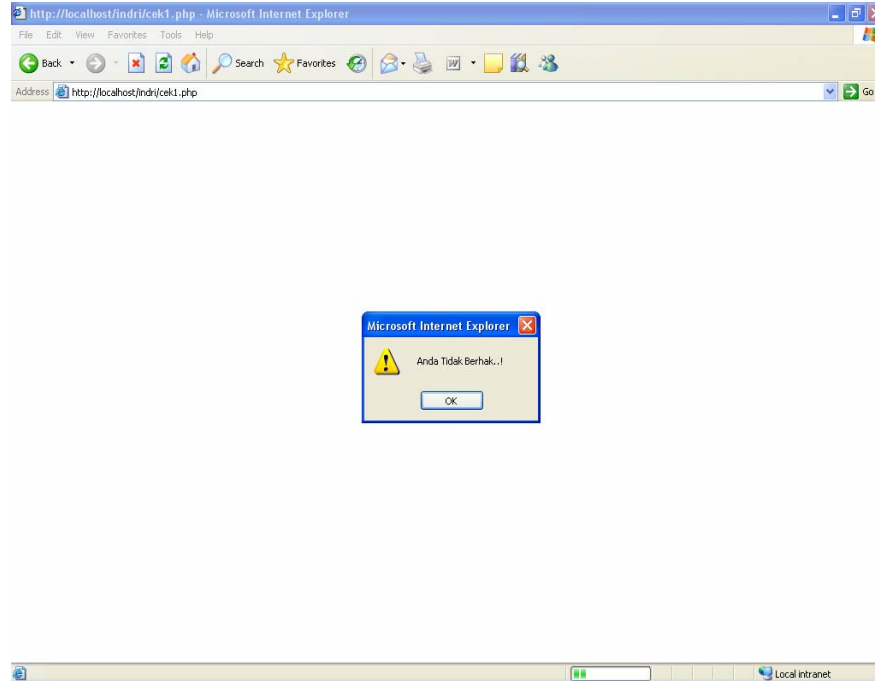
Tabel 4.1 Skenario Pengujian Perangkat lunak

No	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan
1	Login admin	Data salah	Tidak Memasukan <i>username</i> dan <i>password</i>	Menampilkan pesan kesalahan
2	Menu product	Normal	Memilih <i>link</i> produk	Menampilkan halaman produk
3	Menu buku tamu	Data tidak lengkap	Tidak memasukkan alamat <i>e-mail</i>	Menampilkan pesan kesalahan
4	Halaman tambah admin	Data salah	Tidak nama dan password	Menampilkan pesan kesalahan
5	Halaman tambah barang	Data salah	Tidak memasukkan data produk	Menampilkan halaman tambah barang
6	Halaman input report	Normal	Memilih <i>link</i> report	Menampilkan halaman report

4.4.3 Deskripsi Skenario Pengujian

Setelah pengujian perangkat lunak telah dilakukan, selanjutnya melakukan deskripsi skenario dari hasil pengujian perangkat lunak.

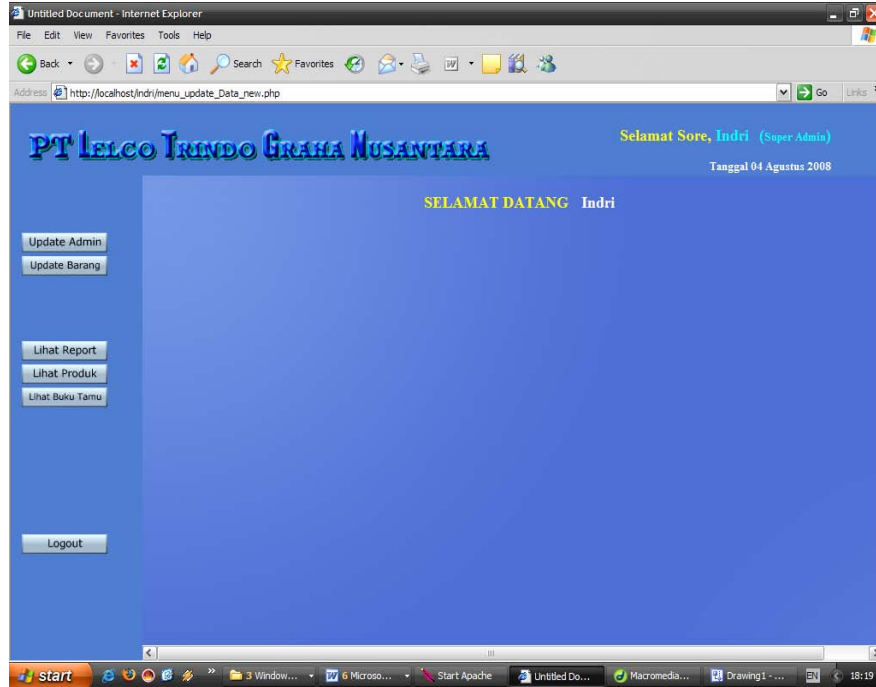
Gambar 4.12 merupakan gambar yang menunjukkan hasil pengujian halaman login admin.



Gambar 4.12 Hasil Pengujian Halaman Login Admin

Pada halaman login admin, pengujian dilakukan dengan kelompok uji salah yang berarti kelompok uji memasukan data salah. Pada Gambar 4.12 kelompok uji tidak memasukkan *username* dan *password*, sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk kembali ke halaman menu administrator. Pengujian pada halaman menu administrator ini menghasilkan rancangan yang sesuai diinginkan.

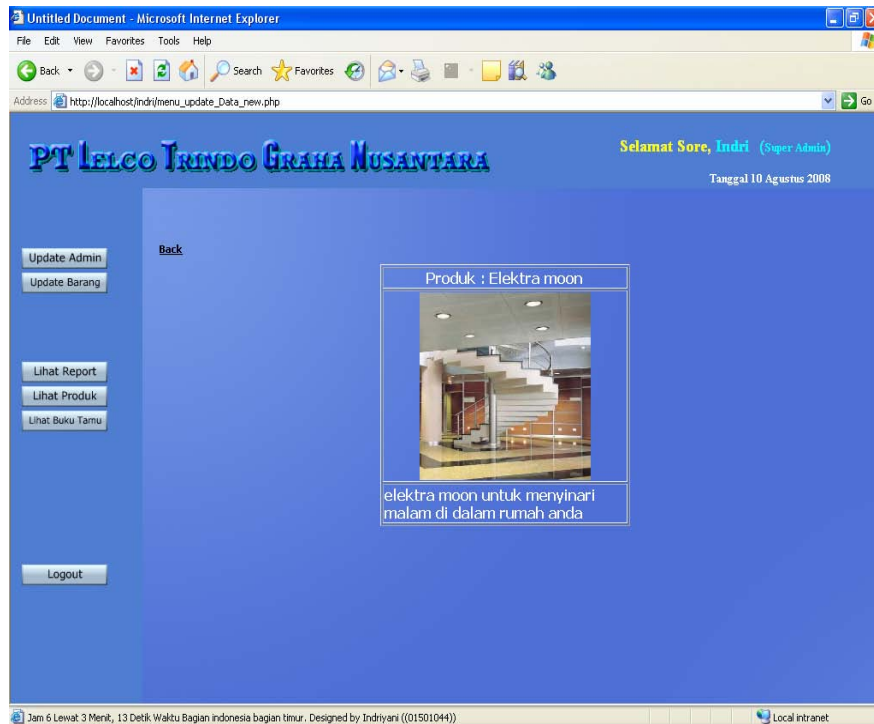
Gambar 4.13 merupakan gambar yang menunjukkan proses yang terjadi setelah administrator melakukan *login*.



Gambar 4.13 Hasil Pengujian Halaman Login administrator

Pada halaman login administrator, pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukan data benar. Kelompok uji melakukan login dengan *username=indri* dan *password=123*, sistem akan memproses username dan password tersebut. Setelah *username* dan *password* diproses, sistem akan menampilkan halaman proses soal, seperti yang terlihat pada Gambar 4.13.

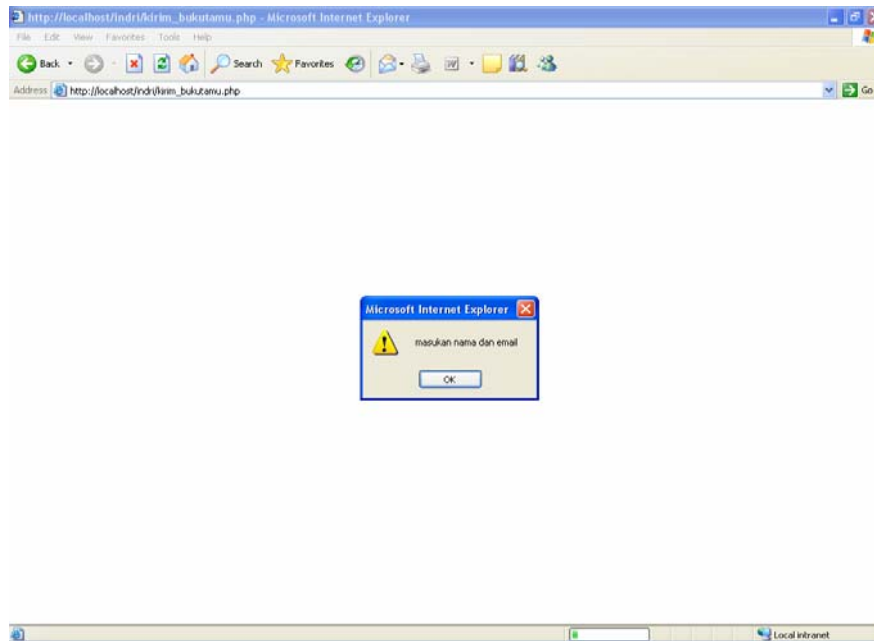
Gambar 4.14 merupakan gambar yang menunjukkan produk perusahaan pada aplikasi.



Gambar 4.14 Hasil Pengujian Halaman produk.

Pada halaman produk, pengujian dilakukan dengan cara memilih menu update barang dan memilih sub menu *detail*. Penambahan produk barang dianggap benar apabila nama, gambar dan keterangan yang di masukkan pada saat proses tambah produk tampil dengan benar.

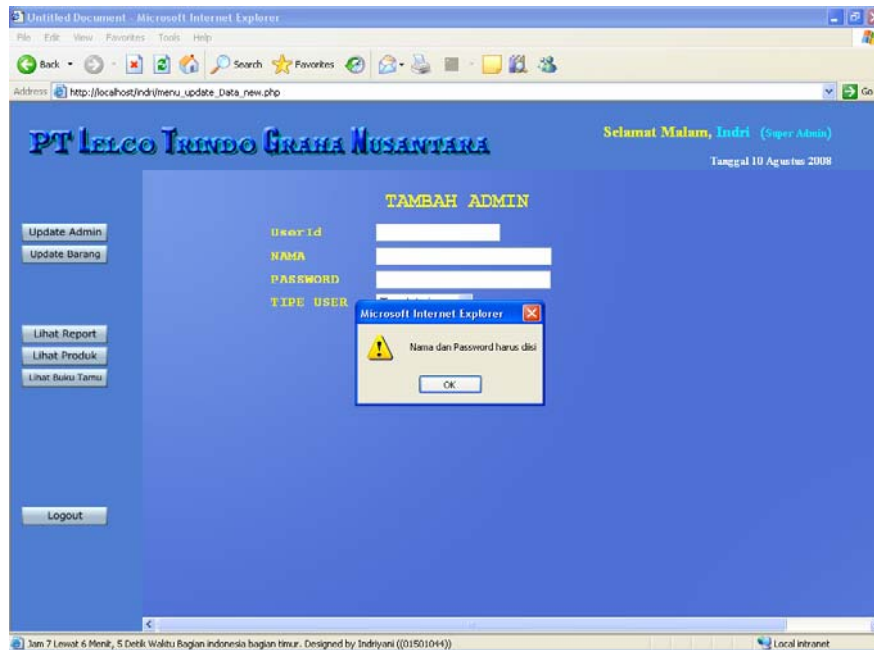
Gambar 4.15 merupakan gambar yang menunjukkan hasil pengujian pada halaman buku tamu.



Gambar 4.15 Hasil Pengujian Halaman menu buku tamu.

Pada halaman menu buku tamu, pengujian dilakukan dengan kelompok uji data salah yang berarti kelompok uji memasukan data salah. Pada Gambar 4.15 kelompok uji tidak memasukan *alamat e-mail* pengunjung. Sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk memasukkan data lengkap yaitu nama dan alamat *e-mail* terlebih dahulu. Rancangan ini berguna untuk mendapatkan data lengkap dari pengirim komentar.

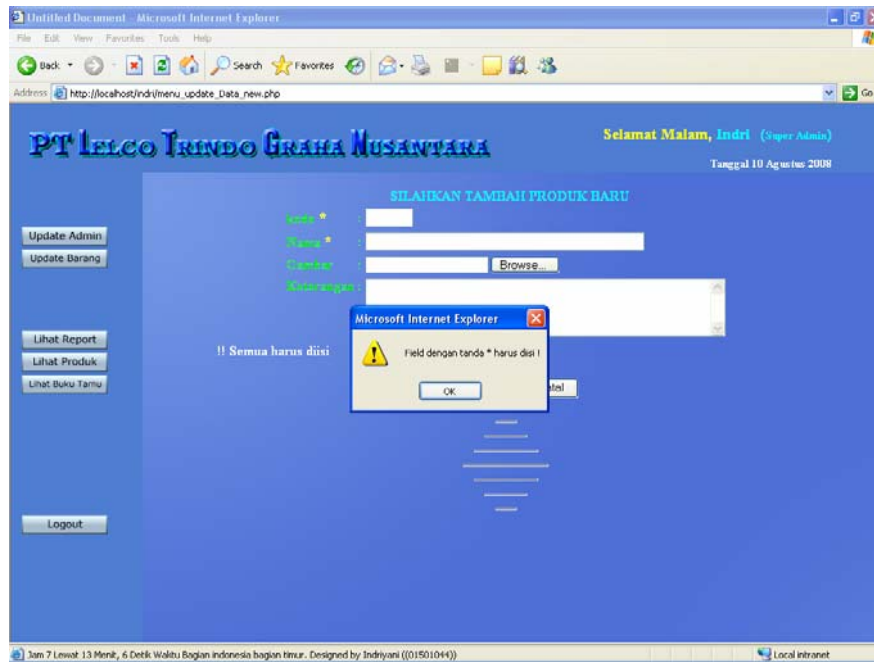
Gambar 4.16 merupakan gambar yang menunjukkan hasil pengujian pada halaman input soal.



Gambar 4.16 Hasil Pengujian Halaman Tambah Admin.

Pada halaman tambah admin, pengujian dilakukan dengan kelompok uji data salah yang dalam pengujian ini kelompok uji tidak memasukkan data. Sistem akan menampilkan pesan kesalahan. Pada halaman ini terdapat tiga buah tombol. Tombol pertama berguna untuk menyimpan data administrator baru ke dalam basis data jika data soal yang dimasukan telah benar, tombol kedua berguna untuk membatalkan pengisian data administrator baru dan tombol *back* untuk mengembalikan tampilan ke menu administrator.

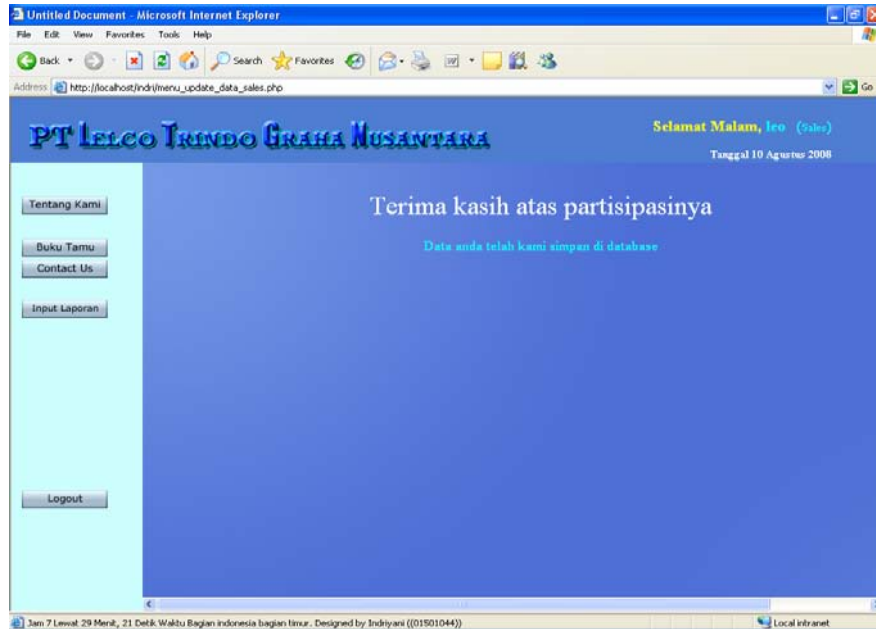
Gambar 4.17 merupakan gambar yang menunjukkan hasil pengujian pada halaman tambah barang.



Gambar 4.17 Hasil Pengujian Halaman Tambah Barang.

Pada halaman tambah barang, pengujian dilakukan dengan kelompok uji data salah yang dalam pengujian ini kelompok uji tidak memasukkan data. Sistem akan menampilkan pesan kesalahan. Pada halaman ini terdapat dua buah tombol. Tombol pertama berguna untuk menyimpan data barang produk baru ke dalam basis data jika data soal yang dimasukan telah benar dan tombol kedua berguna untuk membatalkan pengisian data barang produk baru.

Gambar 4.18 merupakan gambar yang menunjukkan hasil pengujian pada halaman input report.



Gambar 4.18 Hasil Pengujian Halaman input report.

Pada halaman data peserta, pengujian dilakukan dengan kelompok uji benar yang berarti dijalankan dengan semestinya. Pada halaman ini digunakan oleh sales untuk memasukkan data hasil penjualan yang telah diterima oleh sales. Untuk melihat data report yang telah dimasukkan oleh sales, administrator memilih menu update report yang ada pada menu administrator. Sales hanya dapat menginput data sedangkan administrator mempunyai akses untuk merubah dan menghapus laporan yang telah masuk ke basis data perusahaan.

4.4.4 Dokumen Hasil Pengujian

Dokumen hasil pengujian merupakan dokumen yang berisi berdasarkan nomor urut pada skenario pengujian perangkat lunak. Tabel 4.2 adalah tabel dokumen hasil pengujian.

Tabel 4.2 Dokumen hasil pengujian proteksi keamanan sistem informasi user acces.

No	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan	Hasil Uji
1	Login admin	Data salah	Tidak memasukkan username dan password, username="" password=""	Menampilkan pesan kesalahan	Diterima
2	Menu product	Normal	Masuk ke aplikasi memakai username=""indri", password=""123", dan memilih menu update barang	Menampilkan halaman tambah barang	Diterima
3	Menu buku tamu	Data salah	Memasukkan nama = indri, sedangkan alamat e-mail di biarkan kosong	Menampilkan pesan kesalahan	Diterima
4	Halaman tambah admin	Data salah	Menambah data admin dengan mengosongkan field nama="" dan password=""	Menampilkan pesan kesalahan	Diterima
5	Halaman tambah barang	Data salah	pada menu tambah barang dan melakukan proses tambah barang dengan mengosongkan data input barang	Menampilkan pesan kesalahan	Diterima
6	Halaman input report	Normal	Masuk keaplikasi sebagai username=""leo" dan password=""555" dan mengisi data pada input report sales lalu menekan tombol "submit"	Perubahan data disimpan pada basis data dan ditampilkan pada tabel Data report di antarmuka report	Diterima

4.4.5 Analisis Hasil Pengujian

Dari pengujian-pengujian yang telah dilakukan penulis dapat diketahui beberapa hal yaitu:

1. Pada pengujian proses login, untuk berhasil melakukan proses login maka data yang dimasukkan harus benar-benar sama dengan data pada basis data, terutama

yang perlu diperhatikan adalah dari segi perbedaan karakter huruf besar (kapital) dan huruf kecil.

2. Pada pengujian menu produk, data masuk ke aplikasi memakai username, password dan memilih menu update barang. dapat disimpan pada basis data jika data masukan tersebut lengkap dan benar. lengkap disini ditinjau dari segi tidak ada data masukan yang kosong pada field pengisian yang disediakan.
3. Pada pengujian menu buku tamu, jika data yang dimasukkan tidak lengkap yaitu dari segi nama dan alamat *e-mail*. Maka komentar tidak akan disimpan kedalam basis data buku tamu.
4. Pada pengujian menu tambah admin, jika data yang dimasukkan tidak lengkap dan benar benar yaitu dari segi data masukan ada yang bernilai karakter kosong, maka data masukan tersebut tidak akan disimpan di tabel yang bersangkutan pada basis data.
5. Pada pengujian menu tambah barang, perubahan data baru tidak akan dapat dilakukan jika ada data yang dimasukkan tidak lengkap. Jika kondisi tersebut tidak terpenuhi maka proses penambahan barang tidak dapat dilakukan.
6. Hasil yang diperoleh dari pengujian input report, pemasukkan data report hanya dapat dilakukan oleh sales perusahaan. Dan data hanya dapat disimpan pada basis data jika data masukan tersebut lengkap dan benar.

Pengujian-pengujian yang dilakukan oleh penulis terhadap proses-proses di atas menghasilkan proses keluaran sesuai dengan apa yang dirancang dan diharapkan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Ada beberapa hal yang dapat disimpulkan dari uraian pada bab I sampai dengan bab IV. Kesimpulan yang dapat diambil antara lain:

1. Enkripsi menggunakan algoritma MD5 dapat diterapkan pada basis data *user access*. Enkripsi dengan algoritma MD5 dapat dilaksanakan pada panjang data yang variabel. Algoritma ini tidak harus menunggu sejumlah input data tertentu sebelum diproses, atau menambahkan byte tambahan untuk mengenkripsinya.
2. Pengembangan aplikasi keamanan berdasarkan user access pada PT. Lelco Trindo Graha Nusantara ini membutuhkan sebuah basis data dan empat buah tabel yang terdiri dari *tb_admin*, *tb_produk*, *tb_tipeuser*, dan *tb_report*. Semua tabel berguna untuk menyimpan semua data yang diperlukan. Diantara tabel-tabel tersebut terdapat tabel utama, seperti: *tb_admin* dan *tb_tipeuser*. Tabel utama tersebut mempunyai fungsi utama dalam aplikasi ini, yaitu menentukan tingkatan akses.

3. Aplikasi keamanan berdasarkan *user access* yang dibangun telah hampir memenuhi tujuan yang penulis harapkan, yang dalam hal ini pembatasan hak akses dan validasi password menggunakan MD-5 telah berhasil.

5.2 Saran

Tugas Akhir yang dilakukan oleh penulis masih jauh dari sempurna, untuk itu penulis memberikan saran-saran agar dapat dikembangkan, yaitu :

1. Untuk menambahkan keamanan kata kunci (*password*) pada basis data *user access* maka dapat dilakukan enkripsi terlebih dahulu terhadap waktu masuk sebagai kunci dalam algoritma MD5 yang digunakan. Contoh : waktu masuk dienkripsi dengan algoritma MD5 sebagai fungsi hash sehingga panjang kunci setelah dienkripsi menjadi 128 bit.
2. Aplikasi ini merupakan langkah awal dalam rangka mengembangkan teknologi keamanan data pada jaringan berdasarkan *user access*, penulis mengharapkan aplikasi ini dapat dikembangkan agar nantinya menjadi lebih sempurna.

DAFTAR PUSTAKA

- Fathansyah, Ir, 1999 *Basis Data*, Informatika, Bandung
- Komputer, Tim Wahana. 2003. *Memahami Model Enkripsi dan Security Data*. Yogyakarta: Andi.
- Kurniawan Yahya, Jakarta, 2002 *Aplikasi Web Database Dengan PHP Dan MySQL*, Elex Media Komputindo.
- R. Rivest, April 1992 *MIT Laboratory for Computer Science and RSA Data Security, Inc.*
- Sutarman, SKom, 2003 *Membangun Aplikasi Web dengan PHP dan MySQL*, Graha Ilmu.
- http://en.wikipedia.org/wiki/Unified_Modeling_Language, 2008

LAMPIRAN 1 LISTING PROGRAM

1. Listing program antarmuka login administrator

```
<html>
<head>
<title>Login Administrator</title>
</head>
<body bgcolor="#CCCCCC">
<form action="cek1.php"
method="post"
enctype="multipart/form-data" >
<p><br>
<center>
<?
$get=getdate();
$bln=$get['month'];
$tgl=$get['mday'];
$th=$get['year'];
$jam=$get['hours'];
if($jam<=11)
{echo "<font size='+1'
face='tahoma'><b> Selamat Pagi,
</b></font>";}
elseif($jam>11 and $jam <=15)
{echo "<font size='+1'
face='tahoma'><b> Selamat Siang,
</b></font>";}
elseif($jam>15 and $jam <=18)
{echo "<font size='+1'
face='tahoma'><b> Selamat Sore,
</b></font>";}
elseif($jam >18)
{echo "<font size='+1'
face='tahoma'><b> Selamat Malam,
</b></font>";}
echo " <font size='+1'><b>$tgl
$bln $th</b></font> ";
?>
</center><div align="left"><font
size="2"><font face="Times,
serif">USERNAME
</font></div> </td>
<input type="text" name="aman">
</tr>
<div align="left"><font
size="2"><font face="Times,
serif"> PASSWORD</font></div>
<input name="login"
type="password"></font></td></tr>
face="Times New Roman, Times,
serif">
<input type="submit" name="Login"
value="Login">
</font></strong></font></p></div>
</body></html>
<?
if($Login)
{
if($saman =='admin' && $login
=='111')
{
$saman="admin";
$tipe="ADMINISTRATOR";
session_start();
session_register("nama","tipe");
echo
"<script>document.location=\\"menu_
update_admin.php\\"</script>";
exit();
}
else
{
$snama=$saman;
$psw=$login;
include "KONEKSI.php";
$perintah="select * from
ADMINISTRATOR where nama= '$snama'
and password = '$login' " ;
$hasil=mysql_query($perintah) or
die(mysql_error());
$data=mysql_fetch_array($hasil);
if ($data)
{
$nama="$data[nama]";
$tipe="$data[tipeuser]";
session_start();
session_register('nama','tipe');
echo
"<script>document.location=\\"cek2.
php\\"</script>";
exit();
}
else
{echo "<script>alert(\"Anda Tidak
Punya Akses..!\");history.go(-
1)</script>";
exit();}
}
}
?>
```

2. Listing program antarmuka menu utama administrator

```
<?
session_start();
if(!(session_is_registered("tipe")
))
{
echo "<script>alert(\"you're not
licensed to open this
Page!\");history.go(-1)</script>";
exit;
}
?>
<html><head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-
8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
#Layer1 {
position:absolute;
left:10px;
top:68px;
width:155px;
height:31px;
z-index:1;
background-color: #999999;
}
-->
</style></head>
<body bgcolor="#CCCCCC">
<center><? include "top.php"?>
</center>
<table width="882" border="0">
<tr><td width="195">&nbsp;</td>
<td width="146"><form name="form1"
method="post"
action="table_edit_admin.php">
<label></label>
<input type="submit"
name="Submit2" value="Update Data
Karyawan">
</form></td>
<td width="6">&nbsp;</td>
<td width="131"><form name="form2"
method="post"
action="elm_gaji.php">
<label>
<input type="submit" name="Submit"
value="Update Elemen Gaji">
</label>
</form></td>
<td width="3"> </td>
<td width="132"><form name="form5"
method="post"
action="input_gaji.php">
<label>
<input type="submit"
name="Submit5" value="Input Gaji
Bulanan">
</label> </form></td>
<td width="199"><form name="form4"
method="post"
action="view_gaji.php">
<label></label>
<input type="submit"
name="Submit4" value="Lihat Gaji
Karyawan">
</form></td>
<td width="36">&nbsp;</td>
</tr>
</table>
<form name="form3" method="post"
action="logout.php">
<label>
<center><input type="submit"
name="Submit3"
value="Logout"></center>
</label></form></body></html>
```

3. Listing program antarmuka update jabatan

```
<?
session_start();
if(!(session_is_registered("tipe")
))
{
echo "<script>alert(\"Level
Anda Bukan Top
Admin!\");history.go(-1)</script>";
exit;
}
?>
<html><head><title>UPDATE
JABATAN</title>
<script language="JavaScript"
type="text/JavaScript">
<!--
function MM_findObj(n, d) {
//v4.01
var p,i,x; if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.frames.length) {
d=parent.frames[n.substring(p+1)].
document; n=n.substring(0,p);}
if(!(x=d[n])&&d.all) x=d.all[n];
for (i=0;!x&&i<d.forms.length;i++)
x=d.forms[i][n];
for(i=0;!x&&i<d.layers.length;i++)
x=MM_findObj(n,d.layers[i].documen
t);
```



```

    if(!x && d.getElementById)
x=d.getElementById(n); return x;
}
function MM_validateForm() {
//v4.0
    var
i,p,q,nm,test,num,min,max,errors='
',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2);
i+=3) { test=args[i+2];
val=MM_findObj(args[i]);
    if (val) { nm=val.name; if
((val=val.value)!="") {
        if
(test.indexOf('isEmail')!=-1) {
p=val.indexOf('@');
            if (p<1 || p==(val.length-
1)) errors+='- '+nm+' must contain
an e-mail address.\n';
        } else if (test!='R') { num
= parseFloat(val);
            if (isNaN(val)) errors+='-
'+nm+' must contain a number.\n';
        }
        if
(test.indexOf('inRange') != -1) {
p=test.indexOf(':');
            min=test.substring(8,p);
max=test.substring(p+1);
            if (num<min || max<num)
errors+='- '+nm+' must contain a
number between '+min+' and
'+max+'.\n';
        } } } else if (test.charAt(0)
== 'R') errors += '- '+nm+' is
required.\n'; }
    } if (errors) alert('Nama dan
Password harus diisi');
    document.MM_returnValue =
(errors == '');
}
//-->
</script></head><body
bgcolor="#CCCCCC">
<p><center><? include
"top.php"?></center>
<form name="form1" method="post"
action="new_jabatan.php"
enctype="multipart/form-data">
<table width="55%" align="center"
height="45">
<tr>
<td width="2%"><div
align="center"></div></td>
<td width="85%">
<div align="center"><font
color="#000066" size="5"
face="Courier New, Courier,
monospace">
<strong>
<marquee scrollldelay=200>
<font color="#0000FF">UPDATE
JABATAN</font>
</marquee>
</strong></font></div></td>
<td width="13%">&nbsp;  </td>
</tr></table>
<table width="55%" border="1"
cellpadding="1" align="center">
<tr><td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">ID</font></strong></td>
<td width="79%">
<input type="text" name="id"
maxlength="4" size="5"></td></tr>
<tr>
<td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">JABATAN</font></strong>
</td>
<td>
<input type="text" name="jbtn"
size="20"
maxlength="15"></td></tr>
<tr>
<td colspan="2"><div
align="center"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">
<input name="tambah" type="submit"
onClick="MM_validateForm('user','
','R');return
document.MM_returnValue
value="Simpan">
<input name="reset" type="reset"
value="Batal">
</font></strong></div></td></tr>
</table></form>
<form name="form2" method="post"
action="table_edit_admin.php">
<label>
<center><input type="submit"
name="Submit"
value="Back"></center>
</label>
</form><form name="form3"
method="post" action="">
<?
include "KONEKSI.php";
$perintah="SELECT * FROM jabatan
order by id";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
if (empty($data))
{
echo("<font size=3
color=#000000><center><b>DATA
MASIH
KOSONG</center></b></font>");
}
else
{
?>
<table width="415" border="1"
align="center" cellpadding="1"
cellspacing="1">

```

```

<!--DWLayoutTable-->
<tr>
<td width="40" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">ID</font></strong></fon
t></div></td>
<td width="240" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">JABATAN</font></strong>
</font></div></td>
<td width="54" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">HAPUS</font></strong></
font></div></td>
<td width="77" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">EDIT</font></strong></f
ont></div></td></tr>
<?
include "koneksi.php";
$cari="SELECT * FROM jabatan order
by id";
$hasil=mysql_query($cari);
while($data=mysql_fetch_array($has
il))
{
echo "<tr>";
echo("<td align=center><font
color=#000000>$data[id]</font></td
>");
echo strtoupper("<td
align=center><font
color=#000000>$data[jabatan]</font
></td>");
echo("<td align=center><a
href=hapus_jbtn.php?id=$data[id]><
font
color=#000000>hapus</font></a></td
>");
echo("<td align=center><a
href=edit_jbtn.php?id=$data[id]><f
ont
color=#000000>edit</font></a></td>
");
echo("</tr>");
}
}
?>
</table></form></body></html>

```

4. Listing program antarmuka update tipeuser

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{echo "<script>alert(\"Level Anda
Bukan Top Admin\");history.go(-
1)</script>";
exit; }
?>
<html><head><title>UPDATE TIPE
USER</title>
<script language="JavaScript"
type="text/JavaScript">
<!--
function MM_findObj(n, d) {
//v4.01
var p,i,x; if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.fr
ames.length) {
d=parent.frames[n.substring(p+1)].
document; n=n.substring(0,p);}
if(!(x=d[n])&&d.all) x=d.all[n];
for (i=0;!x&&i<d.forms.length;i++)
x=d.forms[i][n];
for(i=0;!x&&d.layers&&i<d.layers.l
ength;i++)
x=MM_findObj(n,d.layers[i].documen
t);
if(!x && d.getElementById)
x=d.getElementById(n); return x; }
function MM_validateForm() {
//v4.0
var
i,p,q,nm,test,num,min,max,errors='
',args=MM_validateForm.arguments;
for (i=0; i<(args.length-2); i+=3)
{ test=args[i+2];
val=MM_findObj(args[i]);
if (val) { nm=val.name; if
((val=val.value)!="") {
if (test.indexOf('isEmail')!=-1) {
p=val.indexOf('@');

```

```

if (p<1 || p==(val.length-1))
errors+="- '+nm+' must contain an
e-mail address.\n";
} else if (test!='R') { num =
parseFloat(val);
if (isNaN(val)) errors+="- '+nm+'
must contain a number.\n";
if (test.indexOf('inRange') != -1)
{ p=test.indexOf(':');
min=test.substring(8,p);
max=test.substring(p+1);
if (num<min || max<num) errors+="-
'+nm+' must contain a number
between '+min+' and '+max+'.\n";
} } } else if (test.charAt(0) ==
'R') errors += '- '+nm+' is
required.\n'; }
} if (errors) alert('Nama dan
Password harus diisi');
document.MM_returnValue = (errors
== ''); }
//-->
</script></head>
<body bgcolor="#CCCCCC">
<p><center><? include
"top.php";?></center>
<form name="form1" method="post"
action="new_tipeuser.php"
enctype="multipart/form-data">
<table width="55%" align="center"
height="45">
<tr><td width="2%"><div
align="center"></div></td>
<td width="85%">
<div align="center"><font
color="#000066" size="5"
face="Courier New, Courier,
monospace">
<strong><marquee scrollldelay=200>
<font color="#0000FF">UPDATE TIPE
USER</font>
</marquee></strong></font></div></td>
<td
width="13%">&nbsp;</td></tr></tabl
e>
<table width="55%" border="1"
cellpadding="1" align="center">
<tr><td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">ID</font></strong></td>
<td width="79%">
<input type="text" name="id"
maxlength="4" size="5"></td></tr>
<tr>
<td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">TIPE
USER</font></strong></td>
<td><input type="text" name="tipe"
size="20"
maxlength="15"></td></tr>
<tr>
<td colspan="2"><div
align="center"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">
<input name="tambah" type="submit"
onClick="MM_validateForm('user','
','R');return
document.MM_returnValue"
value="Simpan">
<input name="reset" type="reset"
value="Batal">
</font></strong></div></td></tr>
</table></form>
<form name="form2" method="post"
action="table_edit_admin.php">
<label>
<center><input type="submit"
name="Submit"
value="Back"></center>
</label></form>
<form name="form3" method="post"
action="">
<?
include "KONEKSI.php";
$perintah="SELECT * FROM TIPEUSER
order by TIPEUSER";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
if (empty($data))
{
echo("<font size=3
color=#000000><center><b>DATA
MASIH
KOSONG</center></b></font>");
}
else
{
?>
<table width="415" border="1"
align="center" cellpadding="1"
cellspacing="1">
<!--DWLayoutTable-->
<tr>
<td width="40" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">ID</font></strong></fon
t></div></td>
<td width="240" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier, monospace">TIPE

```

```

USER</font></strong></font></div><
/td>
<td width="54" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">HAPUS</font></strong></
font></div></td>
<td width="77" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">EDIT</font></strong></f
ont></div></td>
</tr>
<?
include "koneksi.php";
$cari="SELECT * FROM TIPEUSER
order by TIPEUSER";
$hasil=mysql_query($cari);

while($data=mysql_fetch_array($has
il))
{
echo "<tr>";
echo("<td align=center><font
color=#000000>$data[id]</font></td
>");
echo strtoupper("<td
align=center><font
color=#000000>$data[tipeuser]</fon
t></td>");
echo("<td align=center><a
href=hapus_tipe.php?id=$data[id]><
font
color=#000000>hapus</font></a></td
>");
echo("<td align=center><a
href=edit_tipe.php?id=$data[id]><f
ont
color=#000000>edit</font></a></td>
");
echo("</tr>");
}
}
?>
</table></form></body></html>

```

5. Listing program antarmuka input new user

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{echo "<script>alert(\"Anda Tidak
punya hak untuk mengakses halaman
ini\");history.go(-1)</script>";
exit;}
?>
<html><head>
<title>Update      Isi      Tabel
barang</title>
</head>
<body background="newmark1.jpg">
<center> <table width="96%">
<tr>
<td width="48%">
</td>
<td width="52%">
<div align="right">
<object classid="clsid:D27CDB6E-
AE6D-11cf-96B8-444553540000"
codebase="http://download.macromed
ia.com/pub/shockwave/cabs/flash/sw
flash.cab#version=5,0,0,0"
width="100" height="22">
<param name="movie" value="tambah
barang.swf"><param name="quality"
value="high"><param name="bgcolor"
value="#6688E1"><embed src="tambah
barang.swf" quality="high"
pluginspage="http://www.macromedia
.com/shockwave/download/index.cgi?
Pl_Prod_Version=ShockwaveFlash"
type="application/x-shockwave-
flash" width="100" height="22"
bgcolor="#6688E1"></embed>
</object></div></tr></table>
<div align="center"><strong><font
color="#FFFF00" size="5">Table
Barang </font></strong></div>
<table width="100%" border="1"
align="left" cellpadding="1"
cellspacing="1">
<!--DWLayoutTable-->
<tr bgcolor="#00FF00">
<td width="46" height="23"
align="center"><div
align="center"><font
color="#0000FF"><strong><font
size="3" face="Courier New,
Courier,monospace">kode</font></st
rong></font></div></td>
<tdwidth="204" align="center"
valign="top"><div
align="center"><font
color="#0000FF"><strong><font
size="3" face="Courier New,
Courier, monospace">nama barang
</font></strong></font></div></td>
<tdwidth="230"align="center"
valign="top"bgcolor="#00FF00"><div
align="center"><font
color="#0000FF"><strong><font

```

```

size="3" face="Courier New,
Courier, monospace">Gambar</font></
strong></font></div></td>
<td align="center" width="290"><stro
ng><font color="#0000FF" size="3"
face="Courier New, Courier,
monospace">Keterangan</font></stro
ng></td>
<td align="center" width="88"><div
align="center"><font
color="#0000FF"><strong><font
size="3" face="Courier New,
Courier, monospace">edit</font></st
rong></font></div></td>
<td align="center" width="84"><div
align="center"><font
color="#0000FF"><strong><font
size="3" face="Courier New,
Courier, monospace">hapus</font></s
trong></font></div></td></tr>
<?
include "KONEKSI.php";
$cari="SELECT * FROM produk order
by right(kd,length(kd)-1)";
$hasil=mysql_query($cari);
while($data=mysql_fetch_array($has
il)){
echo "<tr>";
echo("<td align=left><font
color=#ffffff><b>$data[kd]</b></fo
nt></td>");
echo("<td align=left><font
color=#ffffff>$data[nama]</font></
td>");
echo("<td align=left><font
color=#ffffff>$data[gambar]</font>
</td>");
echo("<td align=left><font
color=#ffffff>$data[ket]</font></t
d>");
echo("<td align=center><a
href=edit_barang.php?kode=$data[kd
]><font
color=#ffffff>edit</font></a></td>
");
echo("<td align=center ><a
href=hapus_barang.php?kode=$data[k
d]>title='hati-
hati'><fontcolor=#ffffff>hapus</fo
nt></a></td>");
echo("</tr>");
}
?> </table>
<p><b><a href="Menu_Update_Data_new
.php"></a></b></p>
</center></body></html>

```

6. Listing program antarmuka buku tamu

```

<?
session_start();
if (!session_is_registered("nama")
&& !session_is_registered("pass"))
{echo "Anda Tidak Punya akses!!!";
echo "<br>";echo
"<td><b><a href=menu_update_data_ne
w.php>Back</a> </b></td>";
exit;
}
?>
<html><head>
<title>Buku Tamu</title>
</head>
<body background="newmark1.jpg">
<center><div align="center"><font
color="#CC6600"><strong><font
color="#00FFFF" size="5">BuKu Tamu
</font></strong></font></div>
<table width="100%" height="32"
border="1" align="center"
cellpadding="1"
bordercolor="#999999">
<!--DWLayoutTable-->
<tr bgcolor="#00FFFF">
<td width="163" height="26"
align="left" valign="top">
<div align="center"><font
color="#FF0000"><b>Nama</b></font>
</div></td>
<td width="200" align="left"
valign="top">
<div align="center"><font
color="#FF0000"><b>E-
Mail</b></font></div></td>
<td width="509" align="left"
valign="top"><div
align="center"><font
color="#FF0000"><b>Komentar</b></f
ont></div></td>
<td width="86" align="left"
valign="top">
<div align="center"><font
color="#FF0000"><b>hapus</b></font
></div></td>
</tr>
<?
mysql_connect("localhost","","");
mysql_select_db("indri");
$cari="SELECT * FROM bukutamu
order by right(nama,length(nama)-
1)";
$hasil=mysql_query($cari);
while($data=mysql_fetch_array($has
il))
{echo "<tr>";
echo("<td align=left><font
color=#ffffff>$data[nama]</font></
td>");
echo("<td align=left><font
color=#ffffff>$data[email]</font><
/td>");

```

```

echo("<td align=left><font
color=#ffffff>$data[komentar]</font>
</td>");
echo("<td align=left><a
href=hapus_bukutamu.php?nama=$data
[nama]><font

```

```

color=#ffffff><b><center>hapus</center>
</font></a></td>");
echo("</tr>");
?>
</table></center></body></html>

```

7. Listing program antarmuka input buku tamu

```

<html>
<head>
<title>Buku Tamu</title>
</head>
<body
background="newmark1.jpg"><form
name=bukutamu
action=kirim_bukutamu.php
method=post>
<p>&nbsp;</p>
<table width="778" border="0"
cellpadding="0" cellspacing="0">
<!--DWLayoutTable-->
<tr>
<td width="130"
height="39">&nbsp;</td>
<td colspan="6" align="center"
valign="middle"><font size="+2"
face="Comic Strip"><strong>
<marquee scrolldelay="50">
</marquee>
</strong></font><font face="Comic
Strip">&nbsp;</font></td>
<td width="121">&nbsp;</td>
<td width="97">&nbsp;</td>
</tr>
<tr>
<td height="20">&nbsp;</td>
<td width="98">&nbsp;</td>
<td width="118">&nbsp;</td>
<td width="66">&nbsp;</td>
<td width="28">&nbsp;</td>
<td width="58">&nbsp;</td>
<td width="62">&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td height="27">&nbsp;</td>
<td valign="middle"><strong><font
color=#00FF00"
face="Tahoma">Nama</font><font
face="Tahoma"> <font
color=#FF0000">*</font> <font
color=#00FF00">:
</font></strong></td>
<td colspan="6" align="left"
valign="middle"><input name="nama"
type="text" size="50"
maxlength="50">
&nbsp;</td>
</tr>
<tr>
<td height="16">&nbsp;</td>

```

```

<td height="29">&nbsp;</td>
<td
valign="middle"><strong><font
color=#00FF00" face="Tahoma">E-
mail</font><font face="Tahoma">
<font color=#FF0000">*</font>
<font
color=#00FF00">:</font></font></s
trong></td>
<td colspan="6" align="left"
valign="middle"><input
name="email" type="text"
size="50">
&nbsp;</td>
</tr>
<tr>
<td height="24">&nbsp;</td>
<td valign="middle"><strong><font
color=#00FF00"
face="Tahoma">Komentar
:</font></strong></td>
<td colspan="6" rowspan="2"
valign="top"><textarea
name="komen" cols="50"
rows="8"></textarea>
&nbsp;</td>
</tr>
<tr>
<td height="19">&nbsp;</td>
<td>&nbsp;</td>
<td colspan="4" valign="top"><font
color=#FFFF00">Isilah <font
tanda</font> <font
color=#FF0000">*</font> <font
color=#FFFF00">untuk disimpan ke
database</font> </td>
</tr>
<tr>
<td height="24">&nbsp;</td>
<td align="center" valign="middle"><
input name="kirim" type="submit"
value="kirim">&nbsp;</td>
<td>&nbsp;</td>
<td align="center"
valign="middle"><input
type="button"
onClick=location='index.PHP'
value="batal" name="batal">
&nbsp;</td>
</tr>
<tr>
<td height="46">&nbsp;</td>
</tr>
<tr>

```

```
<td height="25"></td>
<td colspan="3" align="center"
valign="middle"><!--
DWLayoutEmptyCell-->&nbsp;</td>
</tr>
<tr>
<td height="9"></td>
</tr>
</table></form>
</body></html>
```

DAFTAR RIWAYAT HIDUP

DATA PRIBADI

Nama : Indriyani
Tempat Tgl. Lahir : Jakarta, 13 Agustus 1983
Jenis Kelamin : Perempuan
Status Perkawinan : Belum Nikah
Agama : Islam
Kewarganegaraan : Indonesia
Alamat : Jl. Lapangan Bola Rt 007/01 No. 29
Kebon Jeruk Jakarta Barat 11530
Tlp. (021) 532 8515 / 081511254677

PENDIDIKAN

No.	Nama Sekolah	Jurusan	Tahun
1	SD Negeri 13 Pagi Jakarta	-	1989-1995
2	SMP Negeri 1 Kerawang	-	1995-1998
3	SMK Ibu Pertiwi Jakarta	-	1998-2001
4	Universitas Mercu Buana, Jakarta	-	2001-2007

PENGALAMAN KEGIATAN AKADEMIK

No.	Kegiatan	Institusi	Tahun
1.	Praktikum Laboratorium : <ul style="list-style-type: none">• Komputer• Bendahara	Lab. Universitas Mercu Buana (Sertifikat) Badan Eksekutif Mahasiswa (BEM) Universitas Mercu Buana	2001 2003

PENGALAMAN KERJA

No.	Aktivitas	Tahun
1.	Karyawan Tata Usaha SDN 10 Pagi Kebon Jeruk	2006
2.	Frontliner & Desk clerk WWW.Voucher_shop.com	2007

Daftar Riwayat Hidup ini dibuat dengan Sebenar-benarnya.

Jakarta, Agustus 2008

Hormat Saya,

(Indriyani)