

II. TINJAUAN PUSTAKA

A. Sistem Informasi Restoran Multi Cabang Manajemen Terpusat

1. Sistem Informasi

Sistem adalah suatu kumpulan komponen yang saling terkait satu sama lain serta berfungsi bersama untuk mencapai suatu tujuan. Sistem mengelola masukan menjadi suatu keluaran. Sedangkan sistem informasi merupakan suatu kumpulan dari perangkat keras, perangkat lunak, data, manusia, dan komponen prosedural untuk menyediakan data dan informasi yang tepat kepada orang yang tepat pada waktu yang tepat [1].

Sedangkan pada definisi lain, Sistem Informasi adalah kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, mengolah tipe transaksi rutin tertentu serta memberi sinyal kepada manajemen yang lainnya terhadap kejadian-kejadian internal [2].

Berdasarkan definisi tersebut dapat disimpulkan bahwa Sistem Informasi adalah kumpulan komponen yang saling terintegrasi untuk mengelola suatu informasi.

Adapun tujuan perancangan sistem informasi antara lain sebagai berikut [3]:

- a. Menyediakan informasi untuk membantu pengambilan keputusan manajemen.
- b. Membantu petugas dalam melaksanakan operasi perusahaan dari hari ke hari.
- c. Menyediakan informasi yang layak untuk pemakai pihak luar perusahaan.

2. Restoran Multi Cabang

Restoran adalah suatu tempat atau bangunan yang diorganisasikan secara komersial yang menyelenggarakan pelayanan dengan baik kepada semua konsumennya baik berupa makanan ataupun minuman [4].

Definisi ‘multi’ berhubungan dengan kata latin *multus* yang berarti ‘banyak’, atau dalam bahasa Yunani sebanding dengan *mala* yang berarti ‘banyak’. Dalam bentuk kombinasinya dengan kata lain, kata ‘multi’ akan menghasilkan makna mempunyai atau terdiri lebih dari satu.

Berdasarkan definisi-definisi tersebut dapat disimpulkan bahwa restoran multi cabang adalah suatu tempat yang menyediakan layanan baik makanan maupun minuman secara komersial kepada konsumennya dan terdiri atas lebih dari satu lokasi atau cabang.

Umumnya sebuah bisnis di bidang restoran berawal dari bisnis yang dikelola oleh keluarga dengan pengelolaan manual yang menuntut adanya rasa percaya terhadap

pengelola suatu cabang. Sistem semacam ini tentunya akan menghambat perkembangan sebuah usaha untuk mendirikan cabang-cabang di berbagai daerah karena keterbatasan sumber daya manusia yang dapat dipercaya dari pihak keluarga. Sebagai contoh, restoran-restoran yang awalnya didirikan atas dasar usaha keluarga di antaranya Dapur Solo, D'Cost, Pecel Lele Lela, dll. Oleh karena itu, sistem informasi diperlukan untuk menjamin pengelolaan sebuah usaha dapat berjalan sebagaimana mestinya serta mampu memberikan laporan hasil usaha secara akurat dan terpercaya kepada pemilik usaha.

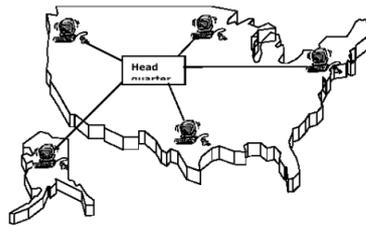
Saat ini perkembangan sistem informasi khususnya di bidang pengelolaan restoran mulai berkembang terutama untuk restoran-restoran berskala besar. Beberapa *software* atau sistem manajemen restoran yang cukup banyak digunakan antara lain Ventrasys, eZee, SolisResto, dll. Restoran-restoran tersebut umumnya menerapkan sistem informasi untuk melakukan pengelolaan dan pemantauan aktivitas bisnis di masing-masing cabang yang mereka miliki. Meskipun sistem informasi memiliki peranan besar dalam membantu manajemen dan operasional sebuah bidang usaha khususnya restoran, namun restoran-restoran dengan banyak cabang di luar provinsi atau pulau bahkan negara saja yang menggunakannya dan belum banyak diikuti oleh restoran-restoran menengah dengan cabang lokal.

3. Manajemen Terpusat

Manajemen adalah kemampuan untuk memonitor, mengontrol dan merencanakan sumber serta komponen sistem. Sedangkan manajemen terpusat merupakan

kemampuan yang digunakan untuk sentralisasi terhadap kontrol dan tanggung jawab manajemen dalam satu lokasi [5].

Manajemen terpusat adalah arsitektur jaringan yang bersandar pada informasi dan kontrol untuk muncul pada sebuah lokasi tunggal yang tersentralisasi atau terpusat. Hal ini menyederhanakan manajemen jaringan yang ukurannya besar. Oleh karena *monitoring* dan kontrol terletak pada satu titik, manajer jaringan harus terletak pada titik yang sama. Supaya bisa memantau jaringan dengan efektif, stasiun manajemen yang terpusat harus menyelidiki peralatan/perlengkapan jaringan yang ditemukan dalam masing- masing jaringan.



Gambar 1. Manajemen terpusat [5].

Sistem manajemen sistem informasi secara terpusat memiliki beberapa manfaat antara lain sebagai berikut :

- a. Kemudahan dalam perencanaan, organisasi, dan pengontrolan kegiatan perusahaan karena tidak perlu terjun langsung ke cabang-cabang yang tersebar di berbagai lokasi untuk melakukan perubahan kebijakan usaha.
- b. Kemudahan dalam memperoleh data-data dari keseluruhan cabang karena tersimpan secara terpusat.
- c. Kemudahan dalam perubahan data karena cabang akan memperbarui data dari pusat.

B. Rekayasa Perangkat Lunak

1. Konsep

Rekayasa perangkat lunak (*software engineering*) berasal dari kata rekayasa (*engineering*) dan perangkat lunak (*software*). Rekayasa merupakan aplikasi terhadap pendekatan sistematis yang berdasar atas ilmu pengetahuan dan matematis serta aplikasi tentang produksi terhadap struktur, mesin, produk, proses atau sistem. Sedangkan perangkat lunak adalah *source code* pada suatu program atau sistem.

Rekayasa Perangkat Lunak (*software engineering*) dapat diartikan dalam beberapa definisi sebagai berikut :

- a. Pembentukan dan penggunaan prinsip rekayasa (*engineering*) untuk mendapatkan perangkat lunak secara ekonomis namun andal dan dapat bekerja secara efisien pada komputer [6].
- b. Penerapan pendekatan yang sistematis, disiplin, dan terukur untuk pengembangan, operasi, dan pemeliharaan perangkat lunak [7].
- c. Suatu disiplin yang mengintegrasikan proses atau prosedur, metode, dan perangkat (*tools*) untuk pembangunan perangkat lunak komputer [8].
- d. Merupakan aplikasi dari prinsip-prinsip *sains* untuk mengurutkan transformasi masalah menjadi solusi yang dapat bekerja dengan baik di mana urutan pemeliharaan perangkat lunak tersebut sampai tidak dapat digunakan lagi [9].

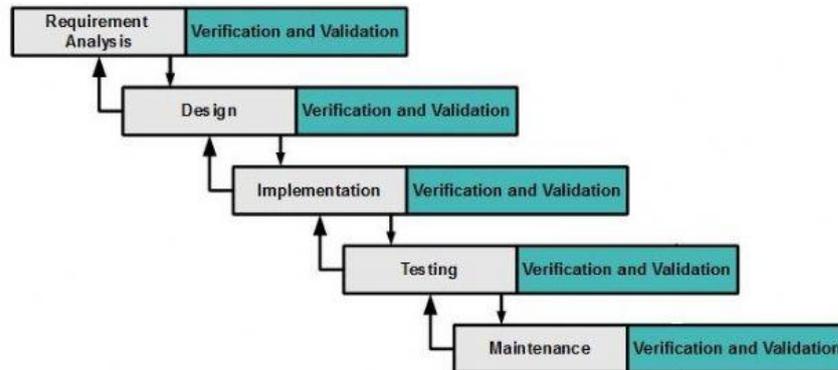
Perangkat lunak mempunyai karakteristik sebagai berikut [10]:

- a. *Maintanability* (dapat dirawat), perangkat lunak harus dapat memenuhi perubahan kebutuhan.
- b. *Dependability*, perangkat lunak harus dapat dipercaya.
- c. Efisiensi, perangkat lunak harus efisien dalam penggunaan sumber daya.
- d. *Usability*, perangkat lunak harus dapat digunakan sesuai dengan yang direncanakan.

2. Model Rekayasa Perangkat Lunak *Modified Waterfall*

Di dalam pengembangan rekayasa perangkat lunak biasanya dipandu dengan pemodelan dengan Daur Hidup Perangkat Lunak (*Software Development Life Cycle*). *Modified Waterfall* adalah salah satu model rekayasa perangkat lunak yang merupakan hasil pengembangan dari model *Classic Waterfall* dengan tahapan-tahapan dan proses pengerjaan yang sama. Perbedaan di antara kedua model tersebut adalah bahwa pada *modified waterfall* memiliki umpan balik (*feedback*) pada setiap tahapnya yang memungkinkan adanya perbaikan pada tahap sebelumnya saat perancangan sedang berlangsung. Hal ini tentu berbeda dengan *classic waterfall* di mana perbaikan hanya dapat dilakukan setelah seluruh tahap perancangan selesai dikerjakan [11].

Adapun pemodelan *Modified Waterfall* dapat diilustrasikan sebagaimana gambar berikut :



Gambar 2. Model *modified waterfall* [11].

Adapun penjabaran dari setiap tahap model *modified waterfall* di atas adalah sebagai berikut :

a. Analisis Kebutuhan (*Requirement Analysis*)

Kebutuhan adalah sesuatu yang disyaratkan, sesuatu yang diinginkan atau diperlukan. Kebutuhan perangkat lunak adalah kondisi, kriteria, syarat atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pemakai [12].

Terdapat tiga macam jenis kebutuhan perangkat lunak [7] :

i. Kebutuhan fungsional (*functional requirement*)

Disebut juga kebutuhan operasional, yaitu kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang harus mampu dikerjakan oleh perangkat lunak.

ii. Kebutuhan antarmuka (*interface requirement*)

Kebutuhan antarmuka yang menghubungkan perangkat lunak dengan elemen perangkat keras, perangkat lunak, atau basis data.

iii. Kebutuhan unjuk kerja (*performance requirement*)

Kebutuhan yang menetapkan karakteristik unjuk kerja yang harus dimiliki oleh perangkat lunak, misalnya kecepatan, ketepatan, frekuensi.

Metode atau teknik untuk melakukan analisis kebutuhan perangkat lunak dapat dikelompokkan berdasarkan pendekatan yang diambil pada saat melakukan aktivitas tersebut. Salah satu metode dalam analisis kebutuhan perangkat lunak adalah Berorientasi Aliran Data (*Data Flow Oriented* atau *Functional Oriented*). Sudut pandang analisis pada pendekatan ini difokuskan pada aspek fungsional dan *behavioral* (perilaku laku) sistem. Pengembang harus mengetahui fungsi-fungsi atau proses-proses apa saja yang ada dalam sistem, data apa yang menjadi masukannya, di mana data tersebut disimpan, transformasi apa yang akan dilakukan terhadap data tersebut, dan apa yang menjadi hasil transformasinya [12].

Adapun perangkat analisis terstruktur antara lain sebagai berikut :

i. Diagram Aliran Data atau *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) merupakan diagram yang menggambarkan aliran data dalam sistem, sumber dan tujuan data, proses pengolahan data tersebut, dan tempat penyimpanan datanya [12]. DFD juga dapat didefinisikan sebagai diagram yang menggunakan notasi simbol untuk menggambarkan arus *data system* [13].

Terdapat tiga macam DFD yang biasa digunakan [14], yaitu :

1. *Context Diagram* (CD)

Yaitu diagram yang paling sederhana dari sebuah sistem informasi untuk menggambarkan aliran-aliran data yang masuk dan keluar pada sistem.

Diagram ini menggambarkan suatu sistem dalam satu lingkaran dan dihubungkan dengan entitas luar yang digambarkan dengan simbol persegi.

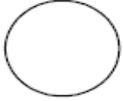
2. DFD Fisik

DFD fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan di mana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan.

3. DFD Logis

DFD logis digunakan untuk membuat dokumentasi sebuah sistem informasi karena DFD logis dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi di mana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan.

Tabel 1. Notasi *Data Flow Diagram* [16].

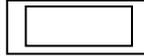
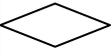
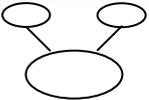
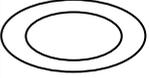
DEMARCO/YOURDON	GANE & SARSON
	
aliran data	aliran data
	
proses	proses
	
entitas eksternal	entitas eksternal
	
data store	data store

ii. Diagram Entitas Relasi atau *Entity Relationship Diagram* (ERD)

ERD adalah diagram yang menggambarkan keterhubungan antar data secara konseptual. Penggambaran keterhubungan antar data ini didasarkan pada anggapan bahwa dunia nyata terdiri atas kumpulan objek yang disebut

entitas (*entity*), dan hubungan yang terjadi di antaranya yang disebut relasi (*relationship*).

Tabel 2. Simbol *Entity Relationship Diagram*.

No	Notasi	Nama	Arti
1		<i>Entity</i>	Objek yang dapat dibedakan dalam dunia nyata
2		<i>Weak Entity</i>	Suatu <i>entity</i> di mana keberadaan dari <i>entity</i> tersebut tergantung dari keberadaan <i>entity</i> yang lain
3		<i>Relationship</i>	Hubungan yang terjadi antara satu atau lebih <i>entity</i>
4		<i>Identifying Relationship</i>	Hubungan yang terjadi antara satu atau lebih <i>weak entity</i>
5		<i>Atribut Simple</i>	Atribut yang bernilai tunggal atau atribut <i>atomic</i> yang tidak dapat dipilah-pilah lagi
6		<i>Atribut Primary Key</i>	Satu atau gabungan dari beberapa atribut yang membedakan semua baris data (<i>row</i>) dalam tabel secara unik
7		<i>Atribut Composite</i>	Atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna
8		<i>Atribut Multivalue</i>	Suatu atribut yang memiliki sekelompok nilai untuk setiap <i>instant entity</i>

b. Perancangan (*Design*)

Perancangan perangkat lunak merupakan tahap kedua dalam model *modified waterfall* dan dapat didefinisikan sebagai berikut :

- i. Proses bertahap di mana semua kebutuhan yang ada diterjemahkan menjadi suatu cetak biru yang akan digunakan untuk mengonstruksi perangkat lunak [8].

- ii. Proses untuk mendefinisikan suatu model atau rancangan perangkat lunak dengan menggunakan teknik dan prinsip tertentu sedemikian hingga model atau rancangan tersebut dapat diwujudkan menjadi perangkat lunak [12].

Adapun tahap-tahap dalam perancangan perangkat lunak antara lain sebagai berikut [12] :

- i. Perancangan data.
- ii. Perancangan arsitektur perangkat lunak.
- iii. Perancangan antarmuka pemakai (*user interface*).
- iv. Perancangan prosedural (spesifikasi program).

c. Implementasi (*Implementation*)

Merupakan sekumpulan aktivitas di mana rancangan perangkat lunak yang telah dibuat pada tahap perancangan kemudian dikodekan ke dalam bentuk kode program dengan menggunakan bahasa pemrograman tertentu agar dapat dijalankan pada komputer [12].

Beberapa faktor standar dalam penyusunan program antara lain [12]:

- i. Kebenaran logika dan penulisan
Program yang disusun harus memiliki kebenaran logika dalam pemecahan masalah maupun penulisan kode program.
- ii. Waktu minimum untuk penulisan program
Penulisan program harus memiliki waktu minimum, artinya waktu minimal yang harus tersedia untuk menuliskan kode program dari awal hingga siap untuk dieksekusi.

iii. Kecepatan maksimum eksekusi program

Agar program memiliki kecepatan eksekusi maksimum, perlu diperhatikan beberapa hal antara lain bahasa pemrograman yang digunakan, algoritma yang disusun, teknik pemrograman yang dipakai, dan perangkat keras yang digunakan.

iv. Ekspresi penggunaan memori

Semakin sedikit penggunaan memori, semakin cepat program dieksekusi.

v. Kemudahan merawat dan mengembangkan program

Program yang memiliki struktur yang baik, struktur data jelas, dan dokumentasi yang lengkap dan mudah dipahami, akan mudah untuk dirawat dan dikembangkan.

vi. *User friendly*

Program yang baik harus memiliki layanan untuk mempermudah pemakai untuk menggunakannya.

vii. Portabilitas

Program yang baik harus dapat dijalankan pada kondisi *platform* yang berbeda-beda, baik itu sistem operasi maupun perangkat keras.

viii. Modular

Pada pendekatan pemrograman, masalah dibagi-bagi menjadi unit terkecil, yang disebut modul untuk menyederhanakan pengimplementasian langkah-langkah pemecahan masalah dalam bentuk program.

d. Pengujian (*Testing*)

Pada proses Rekayasa Perangkat Lunak (RPL), *programmer* mula-mula berusaha untuk membangun perangkat lunak mulai dari konsep abstrak sampai kepada tahap implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

Pada pengujian perangkat lunak, *programmer* menciptakan sekumpulan kasus uji untuk diujikan kepada perangkat lunak. Proses pengujian merupakan tahapan dalam RPL di mana secara fisik terlihat lebih banyak sisi deskruktifnya dibandingkan sisi konstruktifnya karena tujuannya adalah untuk menemukan kesalahan pada perangkat lunak.

Untuk melakukan pengujian perangkat lunak secara fungsional dapat menggunakan metode yang terangkum dalam *Seven-Step Software Testing Process* [15]. Adapun langkah pengujian tersebut adalah sebagai berikut :

i. Organizing for testing

Dalam langkah ini membahas dua hal utama yaitu siapa yang akan terlibat dalam pengujian dan cakupan pengujian.

ii. Developing the test plan

Pada langkah ini pengembang melakukan perancangan terhadap pengujian secara terperinci serta melakukan analisa resiko.

iii. Verification testing

Pada langkah ini dilakukan tiga macam pengujian dasar yaitu kebutuhan, perancangan, dan konstruksi. Pengujian-pengujian tersebut bertujuan untuk meyakinkan bahwa pengembang telah membangun sistem yang benar.

iv. *Validation testing*

Pada langkah ini dilakukan pengujian fungsi dari perangkat lunak yang dibangun secara menyeluruh dan detail untuk mengetahui adanya kesalahan yang masih terjadi serta mencatatnya dalam sebuah laporan pengujian.

v. *Analyzing and reporting test result*

Pada tahap ini merupakan akhir dari pengujian awal di mana selanjutnya pengembang akan melakukan analisa terhadap kesalahan-kesalahan yang tercatat dalam laporan pengujian. Selanjutnya pengembang akan melakukan perbaikan berdasarkan analisa tersebut hingga perangkat lunak siap untuk diujicobakan pada pengguna (*user acceptance test*).

vi. *Acceptance and operational testing*

Berbeda dengan langkah-langkah pengujian sebelumnya di mana pengujian dilakukan selama proses pengembangan oleh pengembang atau tim penguji, pada langkah ini dilakukan pengujian oleh pihak yang akan menggunakan perangkat lunak. Pengujian ini memungkinkan pengguna melakukan evaluasi terhadap pemakaian dan kegunaan dari aplikasi yang telah dibuat dalam aktivitas yang akan mereka lakukan dari hari ke hari. Selain itu juga dilakukan pengujian instalasi pada lingkungan di mana perangkat lunak akan digunakan. Melalui pengujian ini akan meyakinkan pengguna terhadap perangkat lunak ini serta memungkinkan adanya perbaikan atau perubahan sistem.

vii. *Post-implementation analysis*

Merupakan langkah pengujian terakhir yang berisi simpulan tentang hasil akhir pengujian yang dilakukan.

e. Pemeliharaan (*Maintenance*)

Pemeliharaan perangkat lunak merupakan proses memodifikasi sistem perangkat lunak atau komponennya setelah penggunaan oleh konsumen untuk memperbaiki kerusakan, meningkatkan kinerja, manfaat, atau kualitas lainnya atau untuk menyesuaikan sistem perangkat lunak dengan lingkungan yang berubah. Definisi ini menegaskan bahwa proses pemeliharaan perangkat lunak merupakan proses yang bersifat *post-delivery*, artinya dilakukan setelah sistem perangkat lunak digunakan oleh konsumen.

Berikut ini adalah kategori pemeliharaan perangkat lunak :

- i. Pemeliharaan korektif, mencakup semua perubahan yang dilakukan untuk menghilangkan kerusakan aktual pada perangkat lunak.
- ii. Pemeliharaan adaptif, mencakup semua perubahan yang dibutuhkan sebagai konsekuensi dari perubahan lingkungan di mana sistem beroperasi, misalkan perubahan perangkat keras, sistem operasi, DBMS, atau jaringan komputer.
- iii. Pemeliharaan perfektif, mencakup semua perubahan yang berasal dari permintaan pengguna.

C. Sistem Basis Data

1. Konsep Basis Data

Secara umum, basis data (*database*) berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat [16].

Suatu *database* yang baik hendaknya memiliki kriteria sebagai berikut [17] :

- a. *Database* harus bersifat *Object-Oriented* bukan *Program-Oriented*. Maksudnya adalah dalam aplikasi data apabila diakses secara bersamaan, tidak terjadi perubahan pada struktur *database*.
- b. Dapat dikembangkan dalam hal volume dan struktur.
- c. Kerangkapan data (*Data Redundancy*) harus seminimal mungkin.

Dalam konsep pengelolaan basis data dikenal istilah *Data Base Management Systems (DBMS)*. DBMS adalah suatu sistem perangkat lunak yang bisa mendefinisikan, membuat, memelihara, dan mengontrol akses ke Basis Data [18].

2. Manajemen Basis Data dengan MySQL

SQL adalah kependekan dari *Structured Query Language*. Bahasa ini merupakan standar yang digunakan untuk mengakses *database*. Standar SQL mula-mula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*), yang dikenal dengan sebutan SQL86. Saat ini banyak perangkat lunak basis data yang menggunakan SQL sebagai perintah untuk mengakses *database* salah satunya adalah MySQL [16].

MySQL merupakan *software* yang tergolong sebagai DBMS yang bersifat *open source*. MySQL awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia. MySQL diunduh dan digunakan secara gratis karena berlisensi GPL (*General Public License*). Sebagai DBMS, MySQL memiliki sejumlah fitur sebagai berikut :

- a. *Portable*, MySQL dapat berjalan di Sistem Operasi apapun (Windows, MAC dan Linux)

- b. *Freeware*, MySQL seperti halnya php, adalah *freeware* sehingga tidak membutuhkan biaya dalam penggunaannya.
- c. *Multiuser*, MySQL dapat digunakan beberapa pengguna dalam waktu yang serempak tanpa mengalami gangguan.
- d. *Performance turning*, MySQL memiliki kecepatan tinggi dalam memproses *query* biasa. Karena kecepatan MySQL untuk melakukan sebuah *single query* sepuluh kali lebih cepat dibanding dengan PostgreSQL.

D. Pemrograman

Program adalah serangkaian ekspresi yang disusun menjadi kesatuan prosedur berupa urutan langkah untuk menyelesaikan suatu permasalahan dan diimplementasikan dalam bentuk bahasa pemrograman sehingga dapat dijalankan pada komputer. Adapun bahasa pemrograman merupakan tatacara penulisan program [12].

Pada dasarnya, aplikasi dapat dibedakan ke dalam aplikasi berbasis *desktop* dan aplikasi berbasis *website*. Aplikasi Desktop adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi Internet di suatu komputer otonom dengan sistem operasi atau *platform* tertentu. Sedangkan Pengertian aplikasi *web based* adalah suatu aplikasi yang dapat berjalan dengan menggunakan basis teknologi *web* atau *browser*. Aplikasi ini dapat diakses di mana saja asalkan ada koneksi internet yang mendukung, tanpa perlu melakukan instalasi di komputer masing-masing seperti pada aplikasi

desktop, cukup dengan membuka *browser* dan menuju tempat *server* aplikasi tersebut dipasang [19].

1. Pemrograman Berbasis *Desktop* dengan Lazarus

a. Definisi Lazarus

Lazarus adalah sebuah IDE (*Integrated Development Environment*), lingkungan pengembangan perangkat lunak yang terintegrasi sehingga pembuatan *software* dapat diselesaikan dalam waktu singkat. Lazarus bersifat *open source*, tersedia untuk banyak *platform* yakni Linux, Windows dan Mac. Bahasa pemrograman yang dijadikan landasan dalam Lazarus adalah Pascal.

Lazarus menyediakan lingkungan pengembangan mirip dengan Delphi. IDE ini dibangun oleh *Free Pascal Compiler* (FPC) yang mempunyai motto *Write Once Compile Anywhere*, artinya hanya dengan sebuah *source code* program dapat dikompilasi di berbagai *platform* sistem operasi serta berbagai arsitektur *processor* (i386, x86_64, arm, dll) [20].



Gambar 3. Tampilan *screenshot* Lazarus [20].

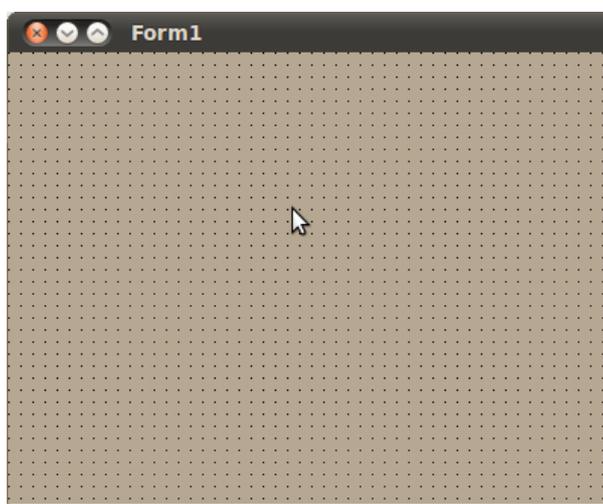
b. IDE (*Integrated Development Environment*) pada Lazarus

Menu-menu penting dalam pemanfaatan Lazarus ditampilkan pada bagian paling atas dari jendela Lazarus. Seperti pada *software* lain, menu File, Edit, Search, View, Window dan Help terdapat pada Lazarus dan dengan mudah dipahami kegunaan dan cara menggunakannya.



Gambar 4. Menu utama Lazarus.

Menu lain yang akan sering digunakan adalah 'Project' dan 'Run'. Menu 'Project' digunakan untuk membuka jendela tertentu untuk mengatur proyek pengembangan *software* yang sedang dilakukan. Dalam Lazarus, setiap program komputer yang dikembangkan dinamakan 'project'. Menu 'Run' digunakan untuk menjalankan, men-*trace* atau menghentikan eksekusi program atau proyek.



Gambar 5. Form Lazarus.

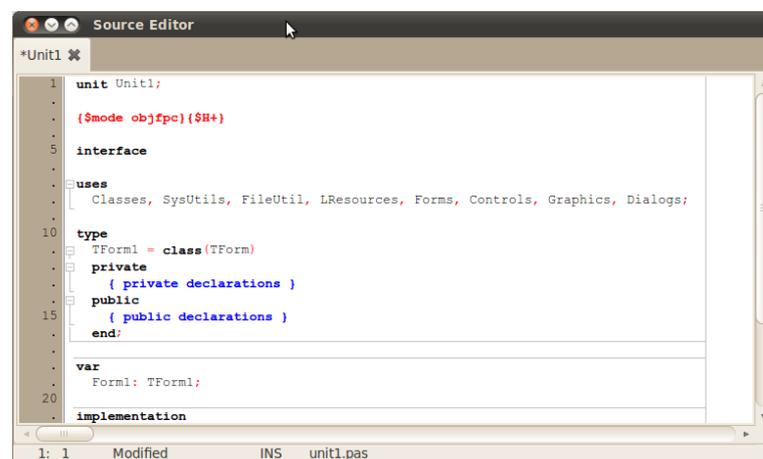
Jendela 'Form' ini adalah bagian utama dari sebuah aplikasi visual. *Form* merupakan komponen paling penting dalam pemrograman visual. Di atas *form* inilah diletakkan komponen-komponen lain.

Pada sisi kiri terdapat sebuah kotak dengan judul ‘*Object Inspector*’. Jendela ini digunakan untuk mengatur perilaku dari suatu kontrol atau komponen yang telah diletakkan di atas suatu *Form* atau jendela aplikasi yang sedang kita kembangkan. Setiap jenis komponen akan mempunyai properti yang berbeda. Properti-properti ini dapat diubah pada saat pengembangan memanfaatkan ‘*object Inspector*’ atau saat eksekusi program dengan menuliskan perintah-perintah dalam kode program.



Gambar 6. *Object Inspector* Lazarus.

Bagian lain yang sangat penting adalah ‘*Source Editor*’. Di sinilah *programmer* menuliskan baris-baris kode program untuk mengatur apa yang harus dilakukan oleh Aplikasi. Kode program ditulis mengikuti aturan dalam bahasa Pascal.



Gambar 7. *Source editor* Lazarus.

c. Konektivitas Lazarus dan MySQL Database

Sangat banyak aplikasi yang dibangun, baik berbasis *Desktop* maupun *Web* memerlukan akses ke *database*. Hal ini tentunya juga menjadi perhatian penting dalam pengembangan *software* dengan Lazarus. Pada dasarnya, Lazarus telah menyediakan dukungan pada berbagai DBMS seperti MySQL 4.x, MySQL 5.x, PostgreSQL, SQLite, SQL Server, Interbase, Firebird dan Oracle [21].

Lazarus memerlukan *library file* 'libmysqlclient.so' agar dapat membangun koneksi ke server *database* MySQL pada Linux OS dan *library file* 'libmysql.dll' untuk Windows OS. *File* tersebut harus berada pada sistem operasi atau berada satu direktori dengan aplikasi yang dijalankan.

Untuk membuat koneksi *database* dengan Lazarus, diperlukan komponen pada *tab* 'SQLdb'. Misalnya untuk membangun koneksi dengan MySQL *database* diperlukan komponen seperti TMySQL50Connection yang digunakan untuk membangun koneksi dengan *database*, TSQLTransaction yang digunakan untuk menerjemahkan *Query* dan TSQLQuery yang digunakan untuk memberikan *Query* pada *database*. Berikut ini adalah contoh kode program untuk melakukan koneksi *database* :

```
MySQL50Connection1.HostName := '127.0.0.1';
MySQL50Connection1.UserName := 'root';
MySQL50Connection1.Password := 'abc123';
MySQL50Connection1.DatabaseName := 'dbstbi';
MySQL50Connection1.Transaction := SQLTransaction1;
SQLQuery1.SQL.Text := 'select * from tbberita';
SQLQuery1.Open;
```

2. Pemrograman Berbasis *Mobile Web* dengan PHP

a. Pemrograman *Web*

Pemrograman *web* diambil dari dua suku kata yaitu pemrograman dan *web*. Pemrograman yang dalam bahasa English adalah *programming* dan diartikan proses, cara, pembuatan program (secara bahasa Indonesia). Sedangkan Definisi *Web* adalah jaringan komputer yang terdiri atas kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui protokol *transfer hypertext*.

Orang banyak mengenal *web* dengan istilah WWW (*world wide web*), *World Wide Web* adalah layanan internet yang paling populer saat ini internet mulai dikenal dan digunakan secara luas setelah adanya layanan WWW . WWW adalah halaman-halaman *website* yang dapat saling terkoneksi satu dengan lainnya (*hyperlink*) yang membentuk samudera belantara informasi. WWW berjalan dengan protokol *HyperText Transfer Protokol* (HTTP).

Halaman *Web* merupakan file teks murni (*plain text*) yang berisi sintaks-sintaks HTML yang dapat dibuka, dilihat maupun diterjemahkan dengan *Internet Browser*. Sintaks HTML mampu memuat konten teks, gambar, audio, video dan animasi. Kini internet identik dengan web, karena kepopuleran web sebagai standar *interface* pada layanan-layanan yang ada di internet, dari awalnya sebagai penyedia informasi, ini digunakan juga untuk komunikasi dari *email* sampai dengan *chatting*, sampai dengan melakukan transaksi bisnis (*commerce*) [22].

b. Mobile Web

Perkembangan teknologi pada perangkat *mobile* membutuhkan perkembangan *software* yang sejalan dengan perkembangan *hardware* pada perangkat *mobile* seperti perangkat *handphone*, *pocket PC*, *console game* dan *multimedia pocket player*. Saat ini pada perangkat *mobile* para pengguna tidak hanya menemukan sebuah aplikasi *standalone* tetapi dengan perangkat *mobile* dapat menggunakan aplikasi-aplikasi *mobile web* sama seperti akses *web* dari sebuah komputer meskipun dengan beberapa keterbatasan.

Pada Mei 2005 *Mobile Web* dikeluarkan oleh W3C dengan nama *Mobile Web Initiative* (MWI) dengan tujuan membuat *web* dapat diakses dari sebuah perangkat *mobile* secara sederhana seperti mengakses *web* dari sebuah komputer *desktop*.

Dalam perancangan *mobile web*, terdapat beberapa hal yang perlu diperhatikan antara lain sebagai berikut [23] :

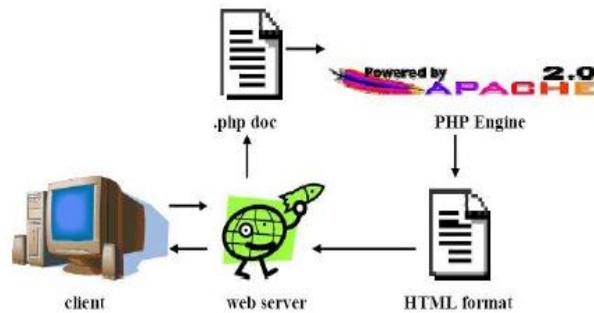
- i. Terdapat beragam *mobile web browser* yang harus didukung oleh *mobile web* yang dirancang.
- ii. Mendukung beragam teknologi *website* terbaru seperti PHP, HTML5, CSS3, dll.
- iii. Perangkat *mobile* cenderung lebih kecil dan lebih lambat sehingga kurang mendukung kompleksitas halaman.
- iv. Tampilan *mobile web* berbeda dengan tampilan *desktop web* sehingga memerlukan perancangan yang berbeda.

c. Bahasa Pemrograman PHP

PHP (Hypertext Preprocessor) merupakan *script* yang dijalankan di *server*, di mana kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi [15]. PHP juga dapat didefinisikan sebagai bahasa pemrograman yang dirancang untuk membuat halaman *web* interaktif dan dijalankan melalui *web server* [24].

PHP adalah produk *open source* yang dapat digunakan secara gratis tanpa harus membayar untuk menggunakannya. Interpreter PHP dalam mengeksekusi kode *PHP* pada sisi *server (server side)*, sedangkan tanpa adanya Interpreter PHP semua *script* dan aplikasi PHP yang dibuat tidak dapat dijalankan. PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang prosesnya secara keseluruhan dijalankan di atas teknologi *web browser* [25].

PHP merupakan aplikasi sisi *server* sehingga proses dilakukan di *server*. Pada saat *browser* meminta dokumen PHP, *web server* langsung menggunakan modul PHP untuk mengolah dokumen tersebut. Jika pada dokumen tersebut terkandung fungsi yang mengakses *database*, maka modul PHP akan menghubungi *database server*. Dokumen PHP akan ditampilkan dalam *web browser* dalam format HTML sehingga kode program PHP tidak tampak di sisi *web browser* [26].



Gambar 8. Pembacaan data pada PHP [26].

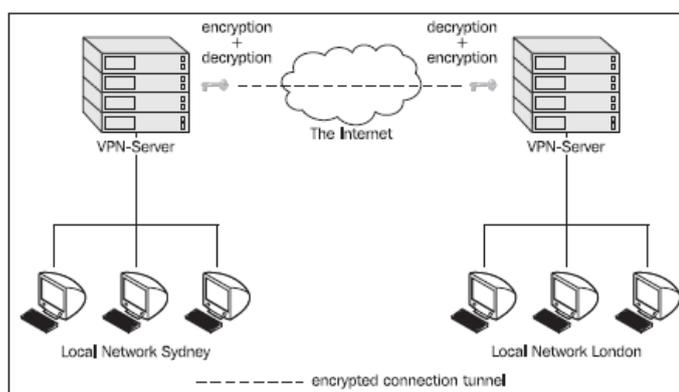
Beberapa kelebihan *PHP* dari bahasa pemrograman *web* lainnya, antara lain sebagai berikut [26] :

1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung *PHP* dapat ditemukan di mana-mana mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. *PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah *system*.

E. VPN

1. Definisi

Virtual Private Network (VPN) adalah sebuah cara mensimulasikan jaringan privat melalui jaringan publik, seperti internet. Dinamakan ‘virtual’ karena menggunakan jaringan virtual atau tanpa adanya koneksi fisik yang menghubungkan jaringan lokal tersebut [27].



Gambar 9. Ilustrasi VPN [27].

2. Kelebihan VPN

Dalam penggunaannya, teknologi VPN menyediakan tiga kelebihan utama [28] sebagai berikut :

a. Kerahasiaan

Teknologi VPN memiliki sistem kerja mengenkripsi semua data yang melewatinya. Dengan adanya teknologi enkripsi ini, kerahasiaan data menjadi lebih terjaga.

b. Integritas Data

Ketika melewati jaringan internet, data sebenarnya sudah berjalan sangat jauh melintasi berbagai negara. Di tengah perjalanannya, apapun bisa terjadi

terhadap isi data tersebut, baik hilang, rusak, maupun dimanipulasi oleh pihak lain. VPN memiliki teknologi enkapsulasi yang dapat menjaga keutuhan data yang dikirim agar mencapai tujuan dengan selamat.

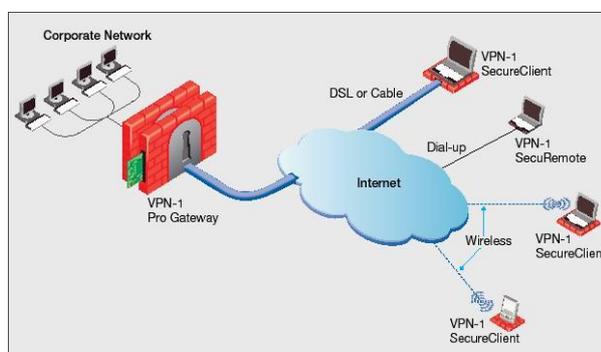
c. Autentikasi Sumber

VPN akan melakukan pemeriksaan terhadap semua data yang masuk dan mengambil informasi sumbernya. Kemudian alamat sumber akan disetujui jika proses autentikasinya berhasil. Dengan demikian, VPN menjamin semua data yang dikirim dan diterima berasal dari sumber yang seharusnya tanpa ada data yang dipalsukan atau dikirimkan oleh pihak asing.

3. Jenis-Jenis VPN

a. *Remote Access VPN*

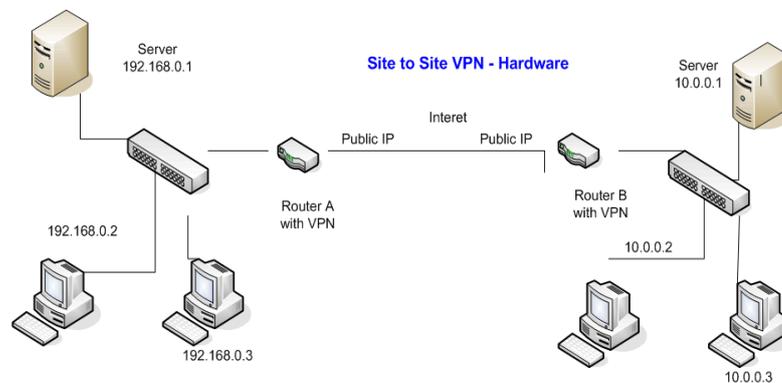
Remote Access VPN memungkinkan akses kapan saja dan di mana saja ke jaringan perusahaan/kantor. Jaringan ini biasa digunakan atau diminta oleh pegawai perusahaan yang berpergian jauh tetapi ingin selalu terhubung dengan jaringan perusahaannya melalui koneksi jarak jauh. Dengan jenis VPN ini memungkinkan kantor cabang melakukan koneksi *dial-up* lokal melalui *provider* dan kemudian melakukan koneksi ke pusat [29].



Gambar 10. *Remote Access VPN* [29].

b. *Site-to-Site* VPN

Site-to-Site VPN memungkinkan suatu *private network* diperluas melintasi jaringan internet atau layanan *public network* lainnya dengan cara yang aman. *Site-to-Site* VPN disebut juga *router-to-router* VPN atau *LAN-to-LAN* dan merupakan salah satu alternatif infrastruktur WAN yang biasa digunakan. VPN jenis ini menghubungkan dua atau lebih kantor cabang, kantor pusat, ataupun partner bisnis ke seluruh jaringan perusahaan [29].



Gambar 11. Site to Site VPN [29].

Site-to-Site VPN terbagi menjadi dua, yaitu:

1. *Intranet* VPN

Intranet VPN digunakan untuk menghubungkan antara kantor pusat dengan kantor cabang atau kantor lain yang letaknya berjauhan melalui jaringan publik.

2. *Extranet* VPN

Extranet VPN digunakan untuk menghubungkan suatu perusahaan dengan perusahaan lainnya (contohnya mitra kerja, pelanggan, atau *supplier*).

4. Prinsip Kerja VPN

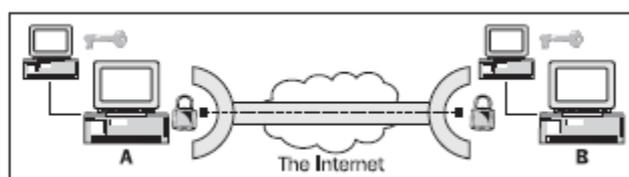
Hal terutama yang dibutuhkan oleh sebuah VPN untuk bekerja adalah adanya koneksi internet yang baik. Kemudian juga diperlukan internet *gateway router* untuk melakukan *setting* akses internet bagi para staf. *Router* ini dikonfigurasi untuk melindungi jaringan lokal perusahaan atau organisasi dari orang yang tidak berhak mengaksesnya melalui internet. Dapat juga dikatakan bahwa *router* ini berfungsi sebagai *firewall*.

Kemudian *VPN software* di-*install* pada *router* yang berfungsi sebagai *firewall* tersebut. Kemudian dikonfigurasi agar dapat tersambung dan tercipta sebuah koneksi *virtual*. Jika tahap ini sukses maka dua atau lebih jaringan perusahaan atau kantor sudah dapat terhubung melalui jaringan *virtual* (internet) layaknya jaringan nyata. Sudah dapat saling mengirim data dan saling mengakses jaringan, namun belum menjadi jaringan *private* karena belum terlindungi, sehingga orang lain yang memakai internet juga dapat mengambil data yang dikirim melalui jaringan ini.

Untuk menjadikan jaringan ini menjadi sebuah jaringan yang *private*, solusinya adalah dengan menggunakan enkripsi. *Traffic* VPN antara dua atau lebih perusahaan/kantor yang menggunakan VPN dikunci dengan enkripsi, dan hanya komputer atau orang yang berhak saja yang dapat membuka kunci dan melihat data yang dikirim dengan enkripsi tersebut. Data yang dikirim akan dienkripsi terlebih dahulu lalu setelah sampai pada tujuan akan didekripsi.

Data dienkapsulasi (dibungkus) dengan *header* yang berisi informasi *routing* untuk mendapatkan koneksi *point to point* sehingga data melewati jaringan publik dan dapat mencapai akhir tujuan. Proses enkapsulasi data sering disebut *tunneling* [29].

Teknologi *Tunnelling* merupakan teknologi yang bertugas untuk menangani dan menyediakan koneksi *point-to-point* dari sumber ke tujuannya. Disebut *tunnel* karena koneksi *point-to-point* tersebut sebenarnya terbentuk dengan melintasi jaringan umum, namun koneksi tersebut tidak mempedulikan paket-paket data milik orang lain yang sama-sama melintasi jaringan umum tersebut, tetapi koneksi hanya melayani transportasi data dari pembuatnya [28].



Gambar 12. *Tunnelling Technology* [28].

5. VPN Software

Selain VPN yang dibangun secara mandiri dengan menggunakan *server* dan kebutuhan tertentu, terdapat pula perusahaan yang memberikan layanan VPN melalui *VIP client software* yaitu aplikasi yang dapat menghubungkan komputer pengguna dengan jaringan atau *server* VPN.

Salah satu *VPN client-software* yang dapat digunakan adalah VPN Hamachi yang merupakan produk dari LogMeIn Inc. yang merupakan sebuah aplikasi *Virtual Private Network* (VPN) yang mampu menghubungkan komputer-komputer

dengan Hamachi di dalamnya dan terkoneksi ke dalam jaringan virtual yang dibangun. Hamachi bersifat *shareware* dengan batasan penggunaan secara bebas untuk maksimal 5 *client* yang terhubung. Hamachi memiliki kemampuan untuk membuat, mengelola, dan memelihara *Virtual Private Network* (VPN) di manapun dengan mudah [30]. Hamachi memiliki beragam pilihan jaringan yang dapat dibangun seperti *Gateway virtual networking*, *Hub-and-spoke virtual networking*, serta jaringan *mesh*.

6. Teknik Pengujian Kualitas Jaringan

a. Perintah PING

Ping (*Packet Internet Gopher*) adalah sebuah program utilitas yang dapat digunakan untuk memeriksa Induktivitas jaringan berbasis teknologi *Transmission Control Protocol/Internet Protocol* (TCP/IP). Dengan menggunakan utilitas ini, dapat diuji apakah sebuah komputer terhubung dengan komputer lainnya. Hal ini dilakukan dengan mengirim sebuah paket kepada alamat IP yang hendak diujicoba konektivitasnya dan menunggu respon darinya [31].

Utilitas Ping akan menunjukkan hasil yang positif jika dua atau buah komputer saling terhubung di dalam sebuah jaringan. Hasil berupa statistik keadaan koneksi kemudian ditampilkan di bagian akhir. Kualitas koneksi dapat dilihat dari besarnya waktu pergi-pulang (*roundtrip*) dan besarnya jumlah paket yang hilang (*packet loss*). Semakin kecil kedua angka tersebut, semakin bagus kualitas koneksinya.

b. Perintah *Trace Route*

Traceroute (Tracert) adalah perintah untuk menunjukkan rute yang dilewati paket untuk mencapai tujuan. Ini dilakukan dengan mengirim pesan *Internet Control Message Protocol (ICMP) Echo Request* ke tujuan dengan nilai *Time to Live* yang semakin meningkat. Rute yang ditampilkan adalah daftar *interface router* (yang paling dekat dengan *host*) yang terdapat pada jalur antara *host* dan tujuan. Jumlah *hop* menunjukkan jumlah *router* atau jaringan luar yang dilewati [32].

c. Pengujian Keamanan dengan *Software Cain & Abel*

Cain & Abel adalah sebuah *software* yang dapat digunakan untuk melakukan penyadapan (*sniffing*) paket jaringan, memecahkan *password* dengan beberapa metode seperti *dictionary attack*, *brute force*, dan *cyptanalysis attack* [33].

d. Penyadapan Paket Jaringan dengan *Software Wireshark*

WireShark adalah suatu perangkat untuk melakukan *Sniffing* paket-paket (*PDU-Protocol Data Unit*) yang berjalan di suatu jaringan yang dapat ditangkap oleh *NIC (Network Interface Card)*. *Wireshark* melakukan penyadapan dengan metode *capture* pada paket-paket yang berjalan pada jaringan untuk dapat dianalisa atau dilihat isinya. Dalam paket tersebut terdapat banyak informasi yang dapat disadap seperti *username*, *password*, atau data penting lainnya [34].