

II. TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah, dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya. [1]

2.2 Komponen dan Elemen Sistem Informasi

Sistem informasi terdiri atas komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri atas komponen input, komponen model, komponen output, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran. [1]

2.2.1 Komponen input

Input mewakili data yang masuk ke dalam sistem informasi. Input di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2.2.2 Komponen model

Komponen ini terdiri atas kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

2.2.3 Komponen output

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

2.2.4 Komponen teknologi

Teknologi merupakan “*tool box*” dalam sistem informasi, Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

2.2.5 Komponen hardware

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung *database* atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

2.2.6 Komponen software

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari *hardware* untuk menciptakan suatu informasi.

2.2.7 Komunikasi data

Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain.

2.3 Arsitektur dan Klasifikasi Sistem Informasi

2.3.1 Arsitektur Sistem Informasi

Sistem informasi dapat dibentuk sesuai kebutuhan organisasi masing-masing. Oleh karena itu, untuk dapat menerapkan sistem yang efektif dan efisien diperlukan perencanaan, pelaksanaan, pengaturan, dan evaluasi sesuai keinginan masing-masing organisasi. Guna dari sistem yang efektif dan efisien tidak lain untuk mendapatkan keunggulan dalam berkompetisi. [1]

Semua orang dapat menggunakan sistem informasi dalam organisasi, tetapi faktor efisiensi setiap sistem adalah berbeda. Perlu diketahui, perubahan sistem, baik besar maupun kecil, selalu akan melalui tingkatan-tingkatan sebagai berikut [1] :

Tabel 2.1 Tingkatan Arsitektur Sistem Informasi

Tingkat I	Ide, mengetahui perlu adanya perubahan.
Tingkat II	Design, merancang cara pemecahannya.
Tingkat III	Pelaksanaan, menerapkan design ke dalam sistem.
Tingkat IV	Kontrol, memeriksa tingkat pelaksanaan dijalankan sesuai dengan design
Tingkat V	Evaluasi, memeriksa apakah perubahan yang terjadi sesuai dengan tujuan semula.
Tingkat VI	Tindak lanjut, melaksanakan perubahan sesuai dengan hasil evaluasi yang ada.

2.3.2 Klasifikasi Sistem Informasi

Sistem informasi merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat di klasifikasikan dari beberapa sudut pandang, diantaranya [1] :

a. Sistem abstrak atau sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem secara fisik, misalnya sistem komputer.

b. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi. Sistem buatan manusia

merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut *human machine system*. Sistem informasi berbasis internet merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem deterministik dan sistem probabilistik

Sistem deterministik adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi. Sistem probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

d. Sistem terbuka dan sistem tertutup

Sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya. Sistem tertutup adalah sistem yang tidak terhubung dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar.

2.4. Bengkel

2.4.1 Pengertian Bengkel

Yang dimaksud dengan bengkel di sini adalah suatu tempat di mana dilakukan perbaikan-perbaikan yang bersifat teknis terhadap suatu produk yang dalam konteks materi ini, produk yang dimaksud adalah kendaraan bermotor. Sebetulnya kegiatan perbengkelan adalah bagian dari kegiatan jaringan layanan purna jual yang sekaligus berfungsi mendukung pemasaran produk yang dijual (yang dalam hal ini adalah kendaraan bermotor). Dalam kenyataannya layanan tidak hanya diberikan kepada kendaraan, tetapi diberikan pula kepada manusianya yaitu

pemilik kendaraan itu sendiri, sehingga mutu pelayanan bagi keduanya harus menjadi perhatian yang serius. [2]

Materi yang diberikan umumnya berfokus kepada perbengkelan kendaraan beroda empat atau lebih, namun masih memungkinkan diaplikasikan untuk kendaraan beroda dua atau lainnya yang juga masih tergolong otomotif. Ada beberapa jenis dan status bengkel yang dapat diterangkan sebagai berikut [2]:

1. Bengkel Bebas (*Independent Work Shop*)

Bengkel ini berdiri sendiri, tidak terikat dan tidak mewakili merek tertentu sehingga kebijakan-kebijakan dapat diambil sendiri sepanjang tidak merugikan bengkel itu sendiri sebagai perusahaan atau sepanjang tidak merusak nama baik perusahaan pemegang merek.

2. Bengkel Perwakilan (*Authorized Work Shop*)

Bengkel ini masih mirip dengan bengkel tersebut di atas, yaitu berdiri sendiri tapi ada merek yang diwakilinya melalui surat penunjukan dari pemegang merek. Kebijakan-kebijakan yang diambil disesuaikan dengan perusahaan yang menunjuknya dan sekaligus masuk ke dalam bagian dari layanan purna jual merek yang bersangkutan. Jenis bengkel ini memungkinkan untuk menerima kemudahan-kemudahan dari perusahaan yang menunjuknya. Kemudahan-kemudahan tersebut bisa bersifat bantuan teknis, permodalan, peralatan atau jenis kemudahan lainnya tergantung dari kebijakan perusahaan yang menunjuknya dan kesepakatan/perjanjian yang dibuat di antara keduanya.

3. Bengkel Dealer (*Dealer Work Shop*)

Bengkel ini merupakan bagian atau sub bagian operasional dari dealer atau ATPM (Agen Tunggal Pemegang Merek) sebagai unit layanan purna jual untuk mendukung sistem pemasaran. Kebijakan-kebijakan yang dibuat sepenuhnya tergantung dan tunduk kepada perusahaan/*dealer* yang bersangkutan.

Bagaimanapun sebuah bengkel adalah sebuah bentuk usaha sehingga secara operasional harus menguntungkan (*Profitable*) dan layak (*Feasible*). Oleh karenanya seluruh kegiatan harus berorientasi kepada perolehan laba. Namun demikian perlu diingatkan bahwa laba harus diperoleh dengan cara yang wajar dan sehat sebab jika tidak demikian justru akan sangat merugikan bengkel itu sendiri karena akan segera ditinggalkan oleh para pelanggannya.[2]

Sumber-sumber penjualan yang dapat dilakukan oleh sebuah bengkel di antaranya adalah sebagai berikut [2]:

1. Penjualan jasa perawatan dan perbaikan (*Maintenance and Repair*)
2. Penjualan suku cadang (*Spare parts*)
3. Penjualan suku cadang tambahan (*Optional parts*)
4. Penjualan barang hiasan (*accessories*)
5. Penjualan minyak pelumas dan minyak hidrolik
6. Penjualan barang-barang lainnya

Adapun ruang lingkup pekerjaan bengkel di antaranya adalah sebagai berikut :

1. Layanan cepat (*Quick service*) dapat berupa, pekerjaan *tune-up*, mengganti minyak pelumas, mencuci dan lain-lain.

2. Perbaikan umum (*General repair*) yang berupa perbaikan *engine*, transmisi, differensial, penyetelan geometrid dan *balancing* roda.
3. Perbaikan elektrik (*Electrical repair*) yang berupa perbaikan system pengapian, starter, pengisian, system penerangan dan instrument.
4. Perbaikan system pendingin ruangan (*Car cooler and Air conditioning*)
5. *Over haul and reconditioning*
6. Perbaikan mesin seperti *boring*, *honing*, bubut rem, skir katup dan lain-lain
7. Perbaikan *body* kendaraan dan cat
8. Perbaikan yang bersifat *fashion* (salon)
9. Pemasangan *accessories* dan *optional parts*
10. Pekerjaan lainnya

Beberapa jenis pekerjaan yang dapat dilakukan sangat tergantung kepada skala bengkel yang harus dipertimbangkan dari banyak hal, misalnya permodalan, jumlah pelanggan, lokasi bengkel, segmen pasar yang diharapkan dan lain-lain.[2]

2.4.2 Konsep Pelayanan Pelanggan Bengkel

Untuk memberikan pelayanan yang baik dibutuhkan kesungguhan yang mengandung unsur kecepatan, keamanan, keramahan, dan kenyamanan yang terintegrasi sehingga manfaatnya besar. Pelayanan adalah setiap kegiatan dan manfaat yang dapat diberikan oleh suatu pihak ke pihak lain yang pada dasarnya tidak berwujud dan tidak perlu berakibat pemilikan sesuatu. Menurut Gasper dalam Mauludin pelayanan didefinisikan sebagai aktivitas pada keterkaitan antara pemasok dan pelanggan untuk memenuhi kebutuhan pelanggan.

Berdasarkan beberapa definisi di atas dapat disimpulkan bahwa pelayanan merupakan suatu tindakan seseorang terhadap orang lain melalui penyajian produk sesuai dengan ukuran berlaku pada produk untuk memenuhi kebutuhan, keinginan dan harapan orang yang dilayani.

Kualitas pelayanan merupakan kunci sukses bagi perusahaan dalam menghadapi era kompetisi yang semakin tajam. Kualitas layanan adalah perbandingan dari harapan pelanggan dengan persepsi dari layanan nyata (*actual performance*) yang mereka terima. Kualitas layanan juga didefinisikan sebagai persepsi konsumen secara keseluruhan baik keunggulan dan kelemahan dari organisasi dalam layanannya.

Menurut Parasuraman ada dua faktor utama yang mempengaruhi kualitas jasa yaitu *expected service* (jasa yang diharapkan) dan *perceived service* (jasa yang diterima). Apabila jasa yang diterima atau yang dirasakan sesuai dengan yang diharapkan, maka kualitas jasa dipersepsikan baik dan memuaskan. Jika jasa yang diterima melampaui harapan pelanggan, maka kualitas jasa dipersepsikan sebagai kualitas yang ideal. Sebaliknya jika jasa yang diterima lebih rendah dari pada yang diharapkan, maka kualitas jasa akan dipersepsikan buruk atau tidak memuaskan. Dengan demikian baik tidaknya kualitas jasa tergantung pada kemampuan pada penyedia jasa dalam memenuhi harapan pelanggan secara konsisten.[22]

dimensi kualitas yang dikembangkan Garvin dalam Tjiptono dapat digunakan sebagai kerangka perencanaan strategis dan analisis. Dimensi-dimensi tersebut antara lain:

1. Kinerja (*performance*) yaitu mengenai karakteristik pokok dari produk inti. Misalnya ketepatan, kemudahan dan kenyamanan.
2. Ciri-ciri atau keistimewaan tambahan (*features*), yaitu karakteristik sekunder atau pelengkap. Misalnya kelengkapan interior dan eksterior.
3. Keandalan (*reliability*), yaitu kemungkinan kecil akan mengalami kerusakan atau gagal dipakai.
4. Kesesuaian dengan spesifikasi (*conformance to specifications*), yaitu sejauh mana karakteristik desain dan operasi memenuhi standar-standar yang telah ditetapkan sebelumnya. Seperti halnya produk atau jasa yang diterima pelanggan harus sesuai bentuk sampai jenisnya dengan kesepakatan bersama.
5. Daya tahan (*durability*), berkaitan dengan berapa lama produk tersebut dapat terus digunakan.
6. *Serviceability*, meliputi kecepatan, kompetensi, kenyamanan, mudah direparasi, serta penanganan keluhan yang memuaskan.
7. Estetika, yaitu daya tarik produk terhadap panca indera, misalnya bentuk fisik motor yang menarik.
8. Kualitas yang dipersepsikan (*perceived quality*), yaitu citra dan reputasi produk serta tanggung jawab perusahaan terhadapnya. Sebagai contoh merek yang lebih dikenal masyarakat (*brand image*) akan lebih dipercaya daripada merek yang masih baru dan belum dikenal. [22]

Menurut Gronroos dalam Tjiptono ada tiga kriteria pokok untuk kualitas pelayanan, yaitu *outcome-related*, *process-related*, dan *image-related criteria*.

Dan ketiga unsur tersebut masih dapat dijabarkan lagi dalam enam dimensi, yaitu:

a. *Professionalism and skills*

Kemampuan, pengetahuan, ketrampilan pada penyedia jasa, karyawan, system operasional, dan sumber daya fisik, dalam memecahkan masalah pelanggan secara professional.

b. *Attitudes and Behavior*

Pelanggan merasa bahwa perusahaan menaruh perhatian dan berusaha untuk membantu dalam memecahkan masalah pelanggan secara spontan dan senang hati.

c. *Accessibility and Flexibility*

Penyediaan pelayanan oleh perusahaan yang dirancang dan dioperasionalkan agar pelanggan mengakses dengan mudah serta bersifat *fleksibel* dalam menyesuaikan permintaan dan keinginan pelanggan.

d. *Reliability and Trustworthiness*

Pelanggan bisa mempercayakan segala sesuatunya kepada penyedia jasa beserta karyawan dan sistemnya.

e. *Recovery*

Proses pengambilan tindakan oleh perusahaan untuk mengendalikan situasi dan mencari pendekatan yang tepat bila pelanggan ada masalah.

f. *Reputation and Credibility*

Keyakinan pelanggan bahwa operasi dari perusahaan dapat dipercaya dan memberikan nilai atau imbalan yang sesuai dengan pengorbanannya.

2.5 SMS Gateway

2.5.1 Pengertian SMS Gateway

Pada dunia komputer, *gateway* dapat diartikan sebagai jembatan penghubung antar satu sistem dengan sistem lain yang berbeda, sehingga dapat terjadi suatu pertukaran data antar sistem tersebut. Dengan demikian, *SMS gateway* dapat diartikan sebagai suatu penghubung untuk lalu lintas data SMS, baik yang dikirimkan maupun yang diterima.[3]

Pada awalnya, *SMS gateway* dibutuhkan untuk menjembatani antar SMSC. Hal ini dikarenakan SMSC yang dibangun oleh perusahaan yang berbeda memiliki protokol komunikasi sendiri, dan protokol-protokol itu sendiri bersifat pribadi. Perhatikan ilustrasi berikut ini.[3]



Gambar 2.1 Jembatan SMSC [3]

Namun seiring perkembangan teknologi komputer, baik dari sisi *hardware* maupun *software*, dan perkembangan teknologi komunikasi, *SMS gateway* tidak lagi dimaksudkan sebagaimana ilustrasi di atas. Dewasa ini, masyarakat lebih mengartikan *SMS gateway* sebagai suatu jembatan komunikasi yang menghubungkan perangkat komunikasi (dalam hal ini ponsel) dengan perangkat komputer, yang menjadikan aktivitas SMS menjadi lebih mudah dan menyenangkan. Pengertian *SMS gateway* kemudian lebih mengarah pada sebuah program yang mengomunikasikan antara sistem operasi komputer, dengan perangkat komunikasi yang terpasang untuk mengirim atau menerima SMS. Salah satu komunikasi yang terjadi, dapat dilakukan dengan mengirimkan perintah AT

pada perangkat komunikasi tersebut, kemudian hasil operasinya dikirimkan kembali ke komputer. [3]



Gambar 2.2 SMS Gateway [3]

2.5.2 Gammu

Gammu adalah nama sebuah proyek yang dikenal dengan nama *command line utility*, yang di mana dapat mengontrol telepon. Ditulis dengan menggunakan bahasa C dan dengan libGammu. Gammu *command line utility* menyediakan akses ke jangkauan yang luas pada fitur-fitur telepon. Adapun fitur-fitur yang di support oleh Gammu ,yaitu [20]

- a. Melakukan Panggilan
- b. Menerima SMS, mengirimnya atau menduplikasikannya
- c. Menerima MMS
- d. Membuat daftar buku telepon, mengeksport atau mengimportnya.

2.6 Sistem Basis Data

2.6.1 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 penganget (*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].

2.6.2 Pengertian MySQL

MySQL merupakan *Software RDMBS* (atau *Server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* dan dapat melakukan suatu proses secara sinkron atau berbarengan.

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar. Lisensi MySQL terbagi menjadi dua. Seorang user dapat menggunakan MySQL sebagai produk *open source* dibawah GNU atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah ke bawah, versi gratis masih dapat digunakan dengan baik. [4]

2.6.3 Kelebihan MySQL

Adapun beberapa alasan kenapa banyaknya *user* yang menggunakan MySQL adalah sebagai berikut : [4]

a. Fleksibel

MySQL dapat digunakan untuk mengembangkan aplikasi *desktop* maupun aplikasi web dengan menggunakan teknologi yang bervariasi. Ini berarti bahwa MySQL memiliki fleksibilitas terhadap teknologi yang akan digunakan sebagai pengembang aplikasi, apakah itu PHP, JSP, Java, Delphi, C++ maupun yang lainnya dengan cara menyediakan *plug-in* dan *driver* yang spesifik untuk masing-masing teknologi tersebut. Dalam database MySQL juga memiliki dukungan terhadap *stored procedure*, fungsi, *trigger*, *view*, SQL standar ANSI, dan lain-lain yang akan mempermudah dan mempercepat proses pengembangan aplikasi.

b. Performa Tinggi

MySQL memiliki mesin *query* dengan performa tinggi, dengan demikian proses transaksional dapat dilakukan dengan sangat cepat. Hal ini terbukti dengan digunakannya MySQL sebagai *database* dari beberapa aplikasi web yang memiliki *traffic* sangat tinggi.

c. Lintas Platform

MySQL dapat digunakan pada *platform* atau lingkungan (dalam hal ini sistem operasi) yang beragam, bisa Microsoft Windows, Linux atau UNIX. Ini menyebabkan proses migrasi data antar sistem operasi dapat dilakukan lebih mudah.

d. Gratis

MySQL dapat digunakan secara gratis. Meskipun demikian, ada juga perangkat lunak MySQL yang bersifat komersial. Biasanya yang sudah ditambah dengan kemampuan spesifik dan mendapat pelayanan dari *technical support*.

e. Proteksi data yang handal

Perlindungan terhadap keamanan data merupakan hal nomor satu yang dilakukan oleh para profesional dalam bidang *database*. MySQL menyediakan mekanisme yang kuat untuk menangani hal tersebut, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.

f. Komunitas Luas

Karena penggunaanya yang banyak, MySQL memiliki komunitas yang luas. Hal ini berguna jika kita menemui suatu permasalahan dalam proses pengolahan data menggunakan MySQL. Dengan mengikuti salah satu atau beberapa komunitas tertentu, kita dapat menanyakan atau mendiskusikan permasalahan tersebut melalui forum. Dengan harapan solusi akan permasalahan tersebut akan cepat diperoleh.

2.7 Rekayasa Perangkat Lunak

2.7.1 Pengertian

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah *Software Engineering*. Istilah *Software Engineering* mulai dipopulerkan tahun 1968 pada *Software Engineering Conference* yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas

pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak dan program komputer.

Perangkat Lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi. Sehingga pengertian RPL sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal, yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. [5]

Adapun pengertian lain menurut pendapat para ahli mengenai Rekayasa Perangkat Lunak, yaitu :

- a. Rekayasa Perangkat Lunak adalah suatu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak, termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak, dan sebagainya. [6]
- b. Suatu disiplin yang mengintegrasikan proses atau prosedur, metode, dan perangkat (*tools*) untuk pembangunan perangkat lunak komputer [7].
- c. Pembentukan dan penggunaan prinsip rekayasa (*engineering*) untuk mendapatkan perangkat lunak secara ekonomis namun andal dan dapat bekerja secara efisien pada 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].

- e. Penerapan pendekatan yang sistematis, disiplin, dan terukur untuk pengembangan, operasi, dan pemeliharaan perangkat lunak [10].

2.7.2 Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain.



Gambar 2.3 Tujuan RPL [5]

Dari gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan *output* yang kinerjanya tinggi, biaya rendah dan waktu penyelesaiannya yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah : [5]

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. menghasilkan perangkat lunak yang kinerjanya tinggi, handal dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah

2.7.3 Ruang Lingkup

Sesuai definisi yang telah disampaikan sebelumnya, ruang lingkup RPL dapat digambarkan sebagai berikut :



Gambar 2.4 Ruang Lingkup [5]

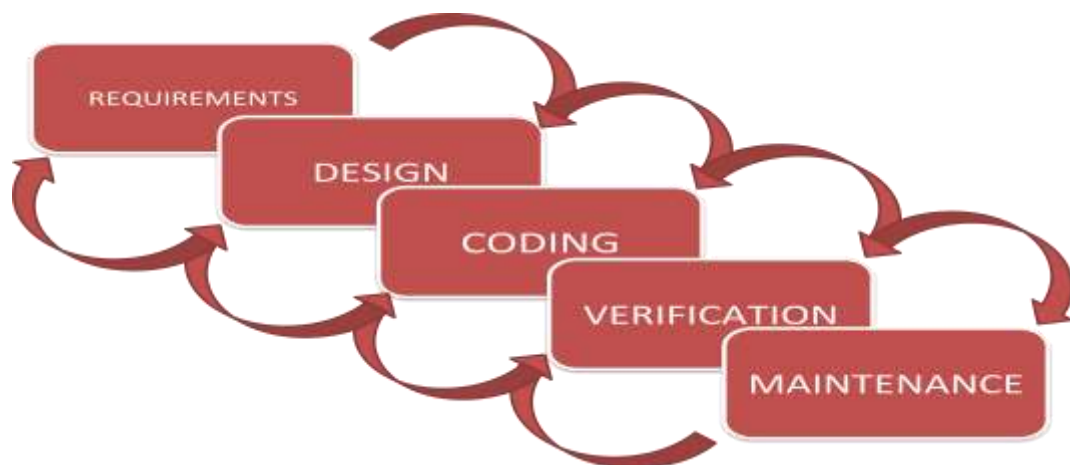
1. *Software Requirements* berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak.
2. *Software Design* mencakup proses penentuan arsitektur, komponen, antarmuka dan karakteristik lain dari perangkat lunak.
3. *Software Construction* berhubungan dengan detail pengembangan perangkat lunak, termasuk algorithman, pengkodean, pengujian dan pencarian kesalahan.
4. *Software Testing* meliputi pengujian pada keseluruhan perilaku perangkat lunak.
5. *Software maintenance* mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
6. *Software configuration management* berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.

7. *Software Engineering management* berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak.
8. *Software Engineering Tools and methods* mencakup kajian teoritis tentang alat bantu dan metode RPL.
9. *Software Engineering process* berhubungan dengan definisi, implementasi, pengukuran, pengelolaan, perubahan dan perbaikan proses RPL.
10. *Software quality* menitikberatkan pada kualitas dan daur hidup perangkat lunak.

2.7.4 Model Modified Waterfall

Model siklus hidup (*life cycle model*) adalah model utama dan dasar dari banyak model. Salah satu model yang cukup dikenal dalam dunia rekayasa perangkat lunak adalah *The Waterfall Model*. Ada 5 tahapan utama dalam *The Waterfall Model*. Disebut *waterfall* (berarti air terjun) karena memang diagram tahapan prosesnya mirip dengan air terjun yang bertingkat.

Tahapan-tahapan dalam *The Waterfall Model* secara ringkas adalah sebagai berikut :



Gambar 2.5 Modified Waterfall

Adapun penjelasan dari setiap tahap-tahap adalah sebagai berikut :

a. Analisis Kebutuhan (*Requirement Analysis*)

Kebutuhan adalah sesuatu yang disyaratkan; sesuatu yang diinginkan atau diperlukan. Sedangkan kebutuhan perangkat lunak adalah kondisi, kriteria, syarat atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pemakai [11].

Secara kategoris, ada tiga buah jenis kebutuhan perangkat lunak [10] :

1. Kebutuhan fungsional (*functional requirement*)

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

2. Kebutuhan antarmuka (*interface requirement*)

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

3. 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 [11].

Adapun perangkat analisis terstruktur antara lain sebagai berikut :

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

Data Flow Diagram (DFD) merupakan diagram untuk menggambarkan aliran data dalam sistem, sumber dan tujuan data, proses yang mengolah data tersebut, dan tempat penyimpanan datanya [11]. DFD juga dapat didefinisikan sebagai diagram yang menggunakan notasi simbol untuk menggambarkan arus *data system*. [12]

Terdapat tiga macam DFD yang biasa digunakan [13], 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

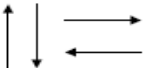
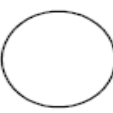
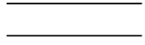
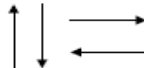
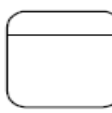
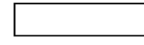
DFD fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan di mana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan.

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 2.2 Notasi DFD [15]

DEMARCO/YOURLDON	GALE & SARSON
   	   

Tabel 2.3 Aturan-aturan dalam DFD [5]

Kelompok	Aturan
Umum	- Masukan ke suatu process akan selalu berbeda dengan keluarannya - Obyek obyek (<i>External Entity</i>, <i>Process</i>, <i>Data Storage</i>, dan <i>Data Flow</i>) yang ada pada suatu DFD selalu memiliki nama yang unik
<i>External Entity</i>	- Nama yang dipakai pada <i>External Entity</i> selalu menggunakan kata benda - Data tidak boleh mengalir secara langsung dari <i>External Entity</i> yang satu ke <i>External Entity</i> yang lain
<i>Process</i>	- Nama yang dipakai pada <i>Process</i> selalu menggunakan kata kerja - Tidak ada <i>Process</i> yang hanya menghasilkan output - Tidak ada <i>Process</i> yang hanya menerima input
<i>Data Storage</i>	- Nama yang dipakai pada <i>Data Storage</i> selalu menggunakan kata benda - Data tidak boleh mengalir secara langsung dari <i>Data Storage</i> yang satu ke <i>Data Storage</i> yang lain - Data tidak boleh mengalir secara langsung dari <i>External Entity</i> ke <i>Data Storage</i> demikian juga sebaliknya.

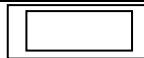
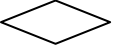
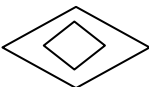
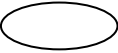
Tabel 2.3 Lanjutan (Aturan-aturan dalam DFD)

<i>Data Flow</i>	• Nama yang dipakai pada <i>Data Flow</i> selalu menggunakan kata benda • <i>Data Flow</i> di antara dua notasi hanya memiliki satu arah aliran • Percabangan (<i>fork</i>) menunjukkan adanya data yang persis sama yang mengalir dari suatu tempat ke dua atau lebih tempat yang lain • Penggabungan (<i>join</i>) menunjukkan adanya data yang persis sama yang mengalir dua atau lebih tempat menuju satu tempat yang lain • <i>Data Flow</i> menuju <i>Data Storage</i> berarti terjadi update data • <i>Data Flow</i> dari <i>Data Storage</i> berarti terjadi pembacaan / pengambilan data
------------------	--

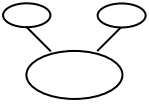
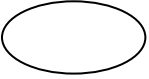
II. Diagram Entitas Relasi atau *Entity Relationship Diagram* (ERD)

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 2.4 Simbol ERD

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

Tabel 2.4 Lanjutan (Simbol ERD) [5]

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. Desain

Desain perangkat lunak adalah tugas, tahapan atau aktivitas yang difokuskan pada spesifikasi detil dari solusi berbasis komputer. Desain perangkat lunak sering juga disebut sebagai *physical design*. Jika tahapan analisis sistem menekankan pada masalah bisnis (*business rule*), maka sebaliknya desain perangkat lunak fokus pada sisi teknis dan implementasi sebuah perangkat lunak.

Output utama dari tahapan desain perangkat lunak adalah spesifikasi desain. Spesifikasi ini meliputi spesifikasi desain umum yang akan disampaikan kepada *stakeholder* sistem dan spesifikasi desain rinci yang akan digunakan pada tahap implementasi. Spesifikasi desain umum hanya berisi gambaran umum agar *stakeholder* sistem mengerti akan seperti apa perangkat lunak yang akan dibangun. Biasanya diagram USD tentang perangkat lunak yang baru merupakan point penting dibagian ini. Spesifikasi desain rinci atau kadang disebut desain arsitektur rinci perangkat lunak diperlukan untuk merancang sistem sehingga memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, bernilai, memiliki aspek *user friendly* dan memiliki dasar-dasar untuk pengembangan selanjutnya.

Desain arsitektur ini terdiri atas desain database, desain proses, desain *user interface* yang mencakup desain *input*, *output form* dan *report*, desain *hardware*, *software* dan jaringan. Desain proses merupakan kelanjutan dari pemodelan proses yang dilakukan pada tahapan analisis. [5]

Adapun tahap-tahap dalam perancangan perangkat lunak antara lain sebagai berikut [11] :

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

c. Implementasi

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 [11].

Beberapa faktor standar dalam penyusunan program antara lain [11]:

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

Pengujian sistem melibatkan semua kelompok pengguna yang telah direncanakan pada tahap sebelumnya. Pengujian tingkat penerimaan terhadap perangkat lunak akan berakhir ketika dirasa semua kelompok pengguna

menyatakan bisa menerima perangkat lunak tersebut berdasarkan kriteria-kriteria yang telah ditetapkan.

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*

Merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan yang terdiri atas : [14]

1. *Corrective Maintenance* : Mengoreksi kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan.

2. *Adaptive Maintenance* : Penyesuaian dengan lingkungan baru, misalnya sistem operasi atau sebagai tuntutan atas perkembangan sistem komputer.
3. *Perfective Maintenance* : Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi – fungsi tambahan, peningkatan kerja dan sebagainya.

2.8 Pemrograman Berbasis *Desktop* dengan Lazarus

2.8.1 Pengertian Lazarus

Lazarus adalah sebuah IDE Delphi yang kompatibel untuk semua *platform* dan dibangun untuk *Free Pascal*. Didalamnya ada LCL yang lebih atau kurang kompatibel dengan VCL dari Delphi. *Free Pascal* adalah Kompiler GPL yang berjalan pada linux, Win32, OS/2, 68K dan masih banyak lagi. *Free Pascal* didesain untuk mampu mengerti dan mengompile *Syntax* Delphi yang termasuk kedalam bagian PBO (Pemrograman berorientasi objek).

Lazarus adalah salah satu bagian dari puzzle yang hilang, Lazarus mengizinkan *programmer* untuk mengembangkan program seperti Delphi di semua *platform* yang disebutkan sebelumnya. Tidak seperti Java yang berusaha untuk ditulis sekali dan dijalankan dimana saja, Lazarus dan *Free Pascal* berusaha untuk ditulis sekali dan dikompile dimana saja. Semenjak kompiler yang tersedia diseluruh *platform*, ini menandakan *programmer* tidak perlu lagi untuk memprogram ulang untuk membangun produk yang identik untuk *platform* yang berbeda. [19].

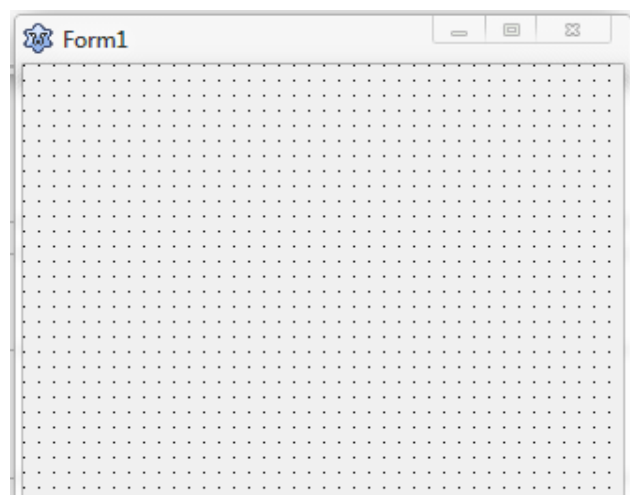
2.8.2 IDE pada Lazarus

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 2.6. 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 2.7. 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.

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;
```

2.9 LAN (*Local Area Network*)

2.9.1 Pengertian LAN

Local Area Network (LAN) adalah sejumlah komputer yang saling dihubungkan bersama di dalam satu areal tertentu yang tidak begitu luas, seperti di dalam satu kantor atau gedung. Secara garis besar terdapat dua tipe jaringan atau LAN, yaitu jaringan *Peer to Peer* dan jaringan *Client-Server*. Pada jaringan *peer to peer*, setiap komputer yang terhubung ke jaringan dapat bertindak baik sebagai *workstation* maupun *server*. Sedangkan pada jaringan *Client-Server*, hanya satu komputer yang bertugas sebagai *server* dan komputer lain berperan sebagai *workstation*. Kedua tipe jaringan tersebut masing-masing memiliki keunggulan dan

kelemahan, di perihal tersebut akan dijelaskan pada tulisan di subbab selanjutnya. LAN tersusun dari beberapa elemen dasar yang meliputi komponen *hardware* dan *software*, yaitu [21] :

a. Komponen Fisik

Personal Computer (PC), *Network Interface Card* (NIC), Kabel,

Topologi Jaringan

b. Komponen Software

Sistem Operasi Jaringan, *Network Adapter Driver*, Protokol Jaringan.

2.9.2 Jaringan *Peer to peer*

Peer To Peer adalah sebuah aplikasi yang mengangani *resource* dari sejumlah *autonomous participant* atau *user* yang terkoneksi secara mandiri, artinya *user* dapat mengoneksikan dirinya sesuai dengan keinginannya, tidak terikat oleh struktur jaringan secara fisik. *Peer-to-peer* menjadi sebuah alternatif aplikasi untuk mencari *resource* tertentu yang tidak ada di *website* ataupun alternatif untuk berbagi *resource* tanpa sebuah *web server* yang harganya masih tergolong mahal. Bila ditinjau dari peran *server* di kedua tipe jaringan tersebut, *server* di jaringan tipe *peer to peer* diistilahkan *non-dedicated server*, karena *server* tidak berperan sebagai *server* murni melainkan sekaligus dapat berperan sebagai *workstation*. [21]

a. Keunggulan

1. Antar komputer dalam jaringan dapat saling berbagi-pakai fasilitas yang dimilikinya seperti : *harddisk, drive, fax/modem, printer*.
2. Biaya operasional relatif lebih murah dibandingkan dengan tipe jaringan *client-server*, salah satunya karena tidak memerlukan adanya *server* yang memiliki kemampuan khusus untuk mengorganisasikan dan menyediakan fasilitas jaringan.
3. Kelangsungan kerja jaringan tidak tergantung pada satu *server*.
Sehingga bila salah satu komputer/*peer* mati atau rusak, jaringan secara keseluruhan tidak akan mengalami gangguan.

b. Kelemahan

1. *Troubleshooting* jaringan relatif lebih sulit, karena pada jaringan tipe *peer to peer* setiap komputer dimungkinkan untuk terlibat dalam komunikasi yang ada. Di jaringan *client-server*, komunikasi adalah antara *server* dengan *workstation*.
2. Unjuk kerja lebih rendah dibandingkan dengan jaringan *client-server*, karena setiap komputer/*peer* di samping harus mengelola pemakaian fasilitas jaringan juga harus mengelola pekerjaan atau aplikasi sendiri.
3. Sistem keamanan jaringan ditentukan oleh masing-masing *user* dengan mengatur keamanan masing-masing fasilitas yang dimiliki. Karena data jaringan tersebar di masing-masing komputer dalam jaringan, maka *backup* harus dilakukan oleh masing-masing komputer tersebut.

2.9.3 Jaringan *Client-Server*

Server adalah komputer yang menyediakan fasilitas bagi komputer-komputer lain didalam jaringan dan *client* adalah komputer-komputer yang menerima atau menggunakan fasilitas yang disediakan oleh *server*. *Server* di jaringan tipe *client-server* disebut dengan *Dedicated Server* karena murni berperan sebagai *server* yang menyediakan fasilitas kepada *workstation* dan *server* tersebut tidak dapat berperan sebagai *workstation*. [21]

a. Keunggulan

1. Kecepatan akses lebih tinggi karena penyediaan fasilitas jaringan dan pengelolaannya dilakukan secara khusus oleh satu komputer (*server*) yang tidak dibebani dengan tugas lain sebagai *workstation*.

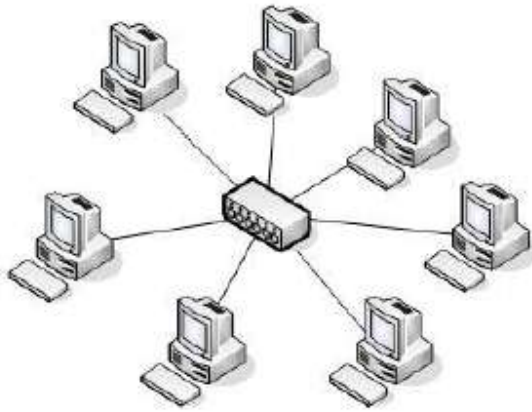
2. Sistem keamanan dan administrasi jaringan lebih baik, karena terdapat seorang pemakai yang bertugas sebagai administrator jaringan, yang mengelola administrasi dan sistem keamanan jaringan.
 3. Sistem backup data lebih baik, karena pada jaringan *client-server* backup dilakukan terpusat di *server*, yang akan mem-*backup* seluruh data yang digunakan di dalam jaringan.
- b. Kelemahan
1. Biaya operasional relatif lebih mahal.
 2. Diperlukan adanya satu komputer khusus yang berkemampuan lebih untuk ditugaskan sebagai *server*.
 3. Kelangsungan jaringan sangat tergantung pada *server* . Bila *server* mengalami gangguan maka secara keseluruhan jaringan akan terganggu.

2.9.4 Topologi *Star* (bintang)

Disebut topologi *star* karena bentuknya seperti bintang, sebuah alat yang disebut *concentrator* bisa berupa *hub* atau *switch* menjadi pusat, di mana semua komputer dalam jaringan dihubungkan ke *concentrator* ini.[21]

1. Pada topologi Bintang (Star) sebuah terminal pusat bertindak sebagai pengatur dan pengendali semua komunikasi yang terjadi. Terminal-terminal lainnya melakukan komunikasi melalui terminal pusat ini.
2. Terminal kontrol pusat bisa berupa sebuah komputer yang difungsikan sebagai pengendali tetapi bisa juga berupa “HUB” atau “MAU” (*Multi Access Unit*).
3. Terdapat dua alternatif untuk operasi simpul pusat.

- i. Simpul pusat beroperasi secara “*broadcast*” yang menyalurkan data ke seluruh arah. Pada operasi ini walaupun secara fisik kelihatan sebagai bintang namun secara logik sebenarnya beroperasi seperti bus. Alternatif ini menggunakan HUB.
 - ii. Simpul pusat beroperasi sebagai “*switch*”, data kiriman diterima oleh simpul kemudian dikirim hanya ke terminal tujuan (bersifat *point-to-point*), alternatif ini menggunakan MAU sebagai pengendali.
- 4. Bila menggunakan HUB maka secara fisik sebenarnya jaringan berbentuk topologi Bintang namun secara logis bertopologi Bus. Bila menggunakan MAU maka baik fisik maupun logis bertopologi Bintang.
 - a. Kelebihan Topologi Bintang
 - 1. Karena setiap komponen dihubungkan langsung ke simpul pusat, pengelolaan menjadi mudah, kegagalan komunikasi mudah ditelusuri.
 - 2. Kegagalan pada satu komponen/terminal tidak mempengaruhi komunikasi terminal lain.
 - b. Kelemahan Topologi Bintang
 - 1. Kegagalan pusat kontrol (simpul pusat) memutuskan semua komunikasi
 - 2. Bila yang digunakan sebagai pusat kontrol adalah HUB maka kecepatan akan berkurang sesuai dengan penambahan komputer, semakin banyak semakin lambat.



Gambar 2.10 Topologi Bintang