BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah Perusahaan

Perkembangan dunia IT yang begitu cepat, memberikan andil yang tidak kecil terhadap segala sudut kehidupan. Era internet pun datang menambah seluruh ruang aktivitas kita. Laju perkembangan IT berimbas pada semakin meningkatnya kebutuhan akan penerapan IT dalam dunia bisnis. Persaingan bisnis yang semakin keras, menuntut pelaku-pelakunya untuk bisa berinovasi menerapkan *tools* pendukung untuk selalu bisa bertahan atau minimal tidak tertinggal terlalu jauh. Salah satu *tools* pendukung itu adalah pemberdayaan teknologi informasi.

PT. Xyclus Cipta Teknologi hadir untuk memenuhi kebutuhan tersebut, PT. Xyclus Cipta Teknologi datang membawa segudang solusi dalam bidang IT. PT. Xyclus Cipta Teknologi adalah perusahaan jasa atau produksi dalam bidang konsultan teknologi informasi, perancangan sistem informasi, jasa integrasi data, pembangunan perangkat lunak, aplikasi berbasis internet dan multimedia.

PT. Xyclus Cipta Teknologi sendiri didirikan pada tahun 2008 di Jalan Bojong Kacor No.1 Cikutra Barat, Bandung. Sampai sekarang, PT. Xyclus Cipta Teknologi telah membangun dan mengembangkan aplikasi perangkat lunak di berbagai bidang seperti Resto, mainan, garmen dan lainnya. Aplikasi yang sudah dikembangkan saat ini sudah berbasis *cloud computing.*

2.1.2 Logo Perusahaan

Berikut merupakan logo dari PT. Xyclus Cipta Teknologi



Gambar 2. 1 *Logo PT. Xyclus Cipta Teknologi*

2.1.3 Visi dan Misi Perusahaan

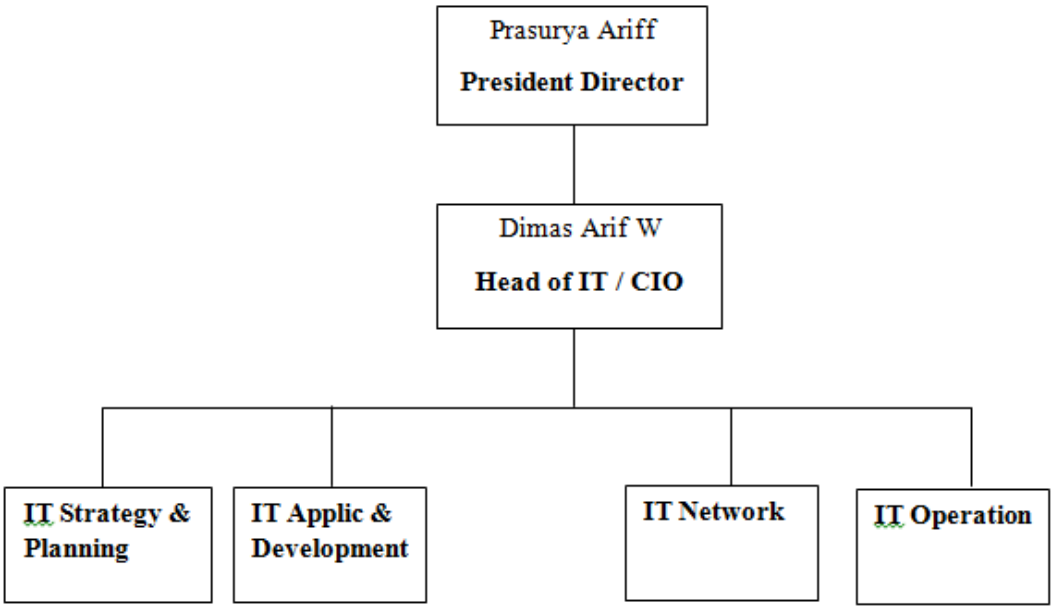
Visi dan misi bagi sebuah perusahaan bisa dikatakan sebagai pedoman dan tujuan. Tanpa adanya visi dan misi sebuah perusahaan tidak akan bertahan dan tidak tahu apa yang harus dilakukan oleh perusahaan tersebut.

Visi : menjadi perusahaan konsultan bisnis dan teknologi informasi yang berorientasi pada pemenuhan kebutuhan pelanggan akan solusi bisnis yang kompetitif dalam menghadapi persaingan global.

Misi : membantu pelanggan dan mitra dalam menciptakan nilai atau daya saing melalui pemanfaatan Teknologi Informasi sehingga memberikan nilai tambah dalam operasional perusahaan maupun pemasaran produk.

2.1.4 Strukur Organisasi Perusahaan

Dapat dilihat pada Gambar 2.2 Struktur Organisasi PT. Xyclus Cipta Teknologi Bandung sebagai berikut:



Gambar 2. 2 *Struktur Organisasi Perusahaan PT. Xyclus Cipta Teknologi*

2.1.5 Tugas dan Wewenang

a. President Director

Presiden direktur adalah pimpinan tertinggi dalam suatu perusahaan . Pimpinan tertinggi ini memiliki tanggung jawab dalam memimpin dan mengarahkan perusahaan, yaitu dalam hal:

1. Menyusun strategi dan visi
2. Menjalin hubungan dan kemitraan strategis
3. Mengatur investasi, alokasi dan divestasi
4. Memimpin direksi

5. Memastikan bahwa prinsip tata kelola perusahaan benar-benar diterapkan dengan baik
6. Membuat rencana pengembangan perusahaan dan usaha perusahaan da lam jangka pendek dan jangka panjang.
7. Bertanggung jawab penuh dalam melaksanakan tugasnya untuk kepen tingan perseroan dalam mencapai maksud dan tujuannya.
8. Menjalin hubungan kerjasama dengan berbagai perusahaan.
9. Tugas dan tanggung jawab President :
10. Bertanggung jawab terhadap kemajuan perusahaan dan mengkoordinasikan selutuh kegiatan perusahaan
11. Memimpin perusahaan dan mengawasi kelancaran perusahaan sesuai dengan tujuan dan kebijakan yang telah ditetapkan
12. Melakukan hubungan dengan pihak luar baik swasta maupun pemerintah yang bertujuan untuk kelancaran perusahaan.
13. Menetapkan kebijakan-kebijakan perusahaan
14. Menerima dan memeriksa laporan dari masing- masing Vice President

b. IT Strategy & Planning

1. Membuat rencana kerja
2. Menugaskan pekerjaan keanggotaan tim
3. Memonitor jalannya proyek dan kerja tim
4. Bertanggung jawan atas keberhasilan eksekusi
5. Memastikan proyek dapat berjalan dengan sukses sesuai dengan jadwal dan ruang lingkup yang disetujui.
6. Mengadakan meeting proyek secara berkala untuk menangani permintaan perubahan.

7. Membuat progress report.
8. Menjembatani penganalisa sistem dengan desainer sehingga antara hasil analisa sistem dan bentuk desain aplikasi sesuai dengan yang diharapkan.
9. Menjembatani desainer dengan programmer sehingga desain baru yang dibuat tidak menyulitkan tim programmer dalam membuat program.

c. IT Applic & Development

IT Applic & Development meliputi System Analis, Programmer, System Desain, Tester dan Dokumentasi

1. System Analyst / Penganalisis Sistem:

1. Melakukan analisis sistem yanga akan dibangun dengan cara wawancara dengan pihak yang ditunjuk.
2. Membuat Statement of Purpose.
3. Membuat Context Diagram.
4. Membuat DFD Level.
5. Membuat Process Specification.
6. Membuat ERD.
7. Membuat Kamus Data.
8. Melaporkan hasil pekerjaan kepada manajer proyek.
9. Berdiskusi dengan manajer proyek mengenai langkah-langkah yang harus ditempuh untuk menyelesaikan proyek yang sesuai dengan perencanaan manajemen proyek (Project management plan).
10. Berdiskusi dengan desainer aplikasi untuk menentukan desain aplikasi sesuai dengan yang diinginkan.
11. Membuat dokumentasi analisa dan desain sistem dibantu oleh dokumentator.

2. Desainer :

1. Membuat desain aplikasi (bentuk menu aplikasi, bentuk form, bentuk report, da).
2. Berdiskusi dengan penganalisis sistem tentang desain aplikasi agar sesuai dengan yang diharapkan.

3. Berdiskusi dengan programmer tentang maksud bentuk desain aplikasi yang dibuat.
4. Membuat dokumentasi desain aplikasi dibantu oleh dokumentator.

3. Programmer :

1. Membuat program aplikasi sesuai dengan hasil analisa dan desain yang ditentukan.
2. Berdiskusi dengan desain aplikasi dan analisis agar program yang dibuatnya sesuai dengan yang telah ditentukan.
3. Memberi usulan pada desainer dan penganalisa sistem (system analyst) bila memang diperlukan.
4. Mengajukan pertanyaan kepada desainer dan system analyst apabila memang ada hal-hal yang kurang jelas.
5. Memperbaiki program berdasarkan hasil investigasi.
6. Membuat dokumentasi program dibantu oleh dokumentator.

4. Tester :

1. Melakukan test terhadap program aplikasi yang telah dibuat.
2. Membuat list tentang hal-hal yang masih perlu diperbaiki.
3. Memberi rekomendasi maupun usulan tentang solusi permasalahan yang dihadapi.
4. Membuat dokumentasi tentang hasil testing terhadap program aplikasi dibantu oleh dokumentator.

5. Dokumentator :

1. Membuat Dokumentasi Sistem (Desain Spesifikasi Sistem), berkoordinasi dengan system analyst, designer, dan programmer.
2. Mendokumentasikan manajemen proyek yang telah dibuat oleh manajer proyek.

d. IT Network

1. Computer Systems Engineers :

1. Berkomunikasi dengan staf atau klien untuk memahami persyaratan sistem tertentu.
2. Memberikan saran pada biaya proyek, konsep desain, atau perubahan desain.
3. Dokumen desain spesifikasi, petunjuk instalasi, dan sistem informasi terkait lainnya.
4. Verifikasi stabilitas, interoperabilitas, portabilitas, keamanan, atau skalabilitas arsitektur sistem.
5. Berkolaborasi dengan engineer atau pengembang perangkat lunak untuk memilih solusi desain yang tepat atau memastikan kompatibilitas komponen sistem.
6. Mengevaluasi teknologi yang muncul saat ini untuk mempertimbangkan faktor-faktor seperti biaya, portabilitas, kompatibilitas, atau kegunaan.
7. Memberikan bimbingan teknis atau dukungan untuk pembangunan atau tips sistem.
8. Mengidentifikasi sistem data, perangkat keras, atau komponen perangkat lunak yang diperlukan untuk memenuhi kebutuhan pengguna.
9. Memberikan pedoman untuk menerapkan sistem yang aman untuk pelanggan atau tim instalasi.
10. Memonitor operasi system untuk mendeteksi masalah potensial.

2. Network and Computer Systems Administrators :

1. Menjaga dan mengelola jaringan komputer dan lingkungan komputasi terkait termasuk perangkat keras komputer, perangkat lunak sistem, perangkat lunak aplikasi, dan semua konfigurasi.
2. Melakukan backup data dan operasi pemulihan kerusakan.
3. Mendiagnosa, memecahkan masalah, dan menyelesaikan perangkat keras, perangkat lunak, atau jaringan lainnya dan masalah sistem, dan mengganti komponen yang rusak bila diperlukan.

4. Merencanakan, mengkoordinasikan, dan melaksanakan langkah-langkah keamanan jaringan untuk melindungi data, perangkat lunak, dan perangkat keras.
5. Mengkonfigurasikan, memonitor, dan memelihara aplikasi email atau virus software perlindungan.
6. Mengoperasikan master konsol untuk memonitor kinerja sistem komputer dan jaringan, dan untuk mengkoordinasikan komputer akses jaringan dan penggunaan.
7. Memuat rekaman komputer dan disk, dan menginstal perangkat lunak dan kertas printer atau form.
8. Desain, mengkonfigurasi, dan perangkat keras uji komputer, jaringan lunak dan perangkat lunak sistem operasi.
9. Memonitor kinerja jaringan untuk menentukan apakah penyesuaian perlu dibuat, dan untuk menentukan di mana perubahan harus dibuat di masa depan.
10. Berunding dengan pengguna jaringan tentang bagaimana untuk memecahkan masalah sistem yang ada.SimakBaca secara fonetik.

3. Computer Security Specialists

1. Mengenkripsi transmisi data dan membangun firewall untuk menyembunyikan informasi rahasia seperti sedang dikirim dan untuk menahan transfer digital tercemar.
2. Mengembangkan rencana untuk melindungi file komputer terhadap modifikasi disengaja atau tidak sah, perusakan, atau pengungkapan dan untuk memenuhi kebutuhan pengolahan data darurat.
3. Meninjau pelanggaran prosedur keamanan komputer dan mendiskusikan prosedur dengan pelanggar untuk memastikan pelanggaran tidak terulang kembali.
4. Memonitor penggunakan file data dan mengatur akses untuk melindungi informasi dalam file komputer.

5. Monitor laporan saat ini dari virus komputer untuk menentukan kapan untuk memperbarui sistem perlindungan virus.
6. Memofifikasi keamanan file komputer untuk memasukkan software baru, memperbaiki kesalahan, atau mengubah status akses individu.
7. Melakukan penilaian risiko dan melaksanakan tes pengolahan data sistem untuk memastikan fungsi pengolahan data kegiatan dan langkah-langkah keamanan.
8. Berunding dengan pengguna untuk membahas isu-isu seperti akses data komputer kebutuhan, pelanggaran keamanan, dan perubahan pemrograman.
9. Melatih pengguna dan meningkatkan kesadaran keamanan untuk memastikan keamanan sistem dan untuk meningkatkan efisiensi server dan jaringan.
10. Mengkoordinasikan pelaksanaan rencana sistem komputer dengan personil pendirian dan vendor luar.

2.2 Landasan Teori

2.2.1 Sistem

2.2.1.1 Pengertian Sistem

Menurut Jogiyanto sistem merupakan bagian yang saling berkaitan erat dan membentuk suatu kesatuan yang saling berinteraksi antara bagian satu dengan bagian lainnya untuk mencapai suatu tujuan, artinya apabila salah satu bagian dari sistem tidak ada maka sistem tersebut tidak akan berfungsi sebagaimana mestinya[2]. Suatu sistem dapat terdiri dari sistem-sistem bagian (subsistem) yang saling berinteraksi, sebagai akibat adanya input yang diproses menjadi output/informasi, misalnya sebuah komputer terdiri dari beberapa komponen.

Pengertian lain sistem menurut Jack Febrian adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu[3].

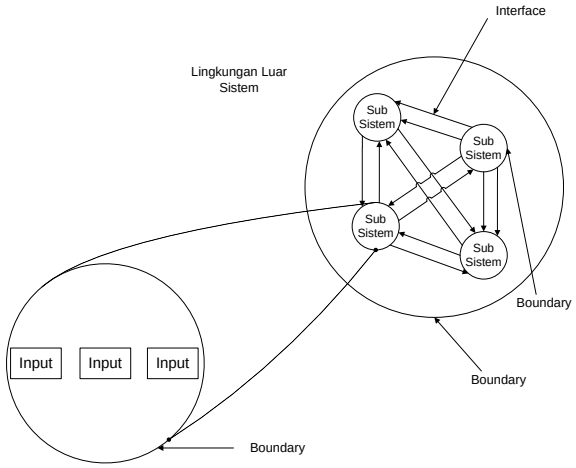
Sedangkan pendekatan sistem yang merupakan jarngan kerja dari prosedur lebih menekankan urut-urutan operasi di dalam sistem. Prosedur (procedure) didefinisikan oleh Richard F. Neuschel yang disadur oleh Jogiyanto mendefinisikan sebagai berikut : "Prosedur adalah suatu urut-urutan operasi klerikal (tulis-menuls) biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis terjadi".

Sedangkan Jerry FitzGerald, Ardra F.FitzGerald dan Warren D.Stallings,Jr. Yang disadur oleh Jogiyanto mendefinisikan prosedur (*procedure*) sebagai berikut: "Suatu prosedur adalah urut-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya".

Dari definisi di atas, maka dapat diambil kesimpulan bahwa sistem merupakan suatu koordinasi dari komponen-komponen yang saling berhubungan dengan tujuan tertentu. Kumpulan dari komponen-komponen yang saling berhubungan dan bertanggung jawab memproses masukan (*input*), sehingga menghasilkan keluaran (*output*) sesuai dengan keinginan. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem yang mana masing-masing menjalankan suatu tertentu yang mempengaruhi proses sistem secara keseluruhan.

2.2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolahan (*process*) dan sasaran (*objective*) atau tujuan (*goal*).



Gambar 2. 3 *Karakterisik Sistem [2]*

2.2.1.3 Kualitas Sistem

Kualitas sistem biasanya berfokus pada karakteristik kinerja sistem. Menurut DeLone dan McLean dalam Livari dan Juhani kualitas sistem merupakan sistem ciri karakteristik kualitas yang diinginkan dari sistem informasi itu sendiri,dan kualitas informasi yang diinginkan informasi karakteristik produk[4]. Kualitas sistem ini juga berarti kombinasi *hardware* dan *software* dalam sistem informasi.

Kualitas sistem memerlukan indikator untuk dapat mengukur seberapa besar kualitas dari sistem. Indikator diperlukan karena kualitas sistem merupakan variabel laten yang tidak dapat diukur secara langsung. Indikator kualitas sistem diwujudkan dalam seperangkat pertanyaan kualitas sistem yang dapat diukur melalui beberapa indikator sebagai berikut:

1. *Ease of Use* (Kemudahan Penggunaan)

Suatu sistem informasi dapat dikatakan berkualitas jika sistem tersebut dirancang untuk memenuhi kepuasan pengguna melalui kemudahan dalam menggunakan sistem informasi tersebut. Davis mengungkapkan kemudahan yang dipersepsikan adalah tingkatan dimana seseorang percaya bahwa pengunaan suatu sistem tertentu dapat menjadikan orang tesebut bebas dari usaha (*free of effort*)[5]. Bebas dari usaha yang dimaksudkan adalah bahwa saat seseorang menggunakan sistem, ia hanya memerlukan sedikit waktu untuk mempelajari sistem tersebut karena sistem tersebut sederhana, tidak rumit, dan mudah dipahami, sudah dikenal (familiar).

Kemudahan penggunaan dalam konteks ini bukan saja kemudahan untuk mempelajari dan menggunakan suatu sistem tetapi juga mengacu pada kemudahan dalam melakukan suatu pekerjaan atau tugas dimana pemakaian suatu sistem akansemakin memudahkan seseorang dalam bekerja dibanding mengerjakan secara manual. Pengguna sistem informasi mempercayai bahwa sistem informasi yang lebih fleksibel, mudah dipahami dan mudah pengoperasiannya sebagai karakteristik kemudahan penggunaan.

2. *Response Time* (Kecepatan Akses)

Kecepatan akses merupakan salah satu indikator kualitas sistem informasi. Jika akses sistem informasi memiliki kecepatan yang optimal maka layak dikatakan bahwa sistem informasi yang diterapkan memiliki kualitas yang baik. Kecepatan akses akan meningkatkan kepuasan pengguna dalam menggunakan sistem informasi.

3. *Reliability* (Keandalan Sistem)

Sistem informasi yang berkualitas adalah sistem informasi yang dapat diandalkan. Jika sistem tersebut dapat diandalkan maka sistem informasi tersebut layak digunakan. Keandalan sistem informasi dalam konteks ini adalah ketahanan

sistem informasi dari kerusakan dan kesalahan. Keandalan sistem informasi ini juga dapat dilihat dari sistem informasi yang melayani kebutuhan pengguna tanpa adanya masalah yang dapat mengganggu kenyamanan pengguna dalam menggunakan sistem informasi.

4. *Flexibility* (Fleksibilitas)

Fleksibilitas suatu sistem informasi menunjukkan bahwa sistem informasi yang diterapkan tersebut memiliki kualitas yang baik. Fleksibilitas yang dimaksud adalah kemampuan sistem informasi dalam melakukan perubahan-perubahan kaitannya dengan memenuhi kebutuhan pengguna. Pengguna akan merasa lebih puas menggunakan suatu sistem informasi jika sistem tersebut fleksibel dalam memenuhi kebutuhan pengguna.

5. *Security* (Keamanan)

Suatu sistem informasi dapat dikatakan baik jika keamanan sistem tersebut dapat diandalkan. Keamanan sistem ini dapat dilihat melalui data pengguna yang aman disimpan oleh suatu sistem informasi. Data pengguna ini harus terjaga kerahasiaannya dengan cara data disimpan oleh sistem informasi sehingga pihak lain tidak dapat mengakses data pengguna secara bebas. Jika data pengguna dapat disimpan secara aman maka akan memperkecil kesempatan pihak lain untuk menyalahgunakan data pengguna sistem informasi.

Menurut Guimares [15], faktor yang digunakan dalam mengukur sebuah kualitas sistem, adalah:

a) *Ease of Use*, apakah sistem mudah untuk digunakan

b) *Ease of Learning*, apakah sistem rentan waktu untuk dipelajari bagi pengguna-pengguna baru

c) *Flexibility*, informasi dapat digunakan untuk berbagai tujuan lain

d) *Sophistication*, informasi memiliki tampilan yang baik

e) *Customization*, pengaturan ulang yang dilakukan oleh user pada software yang bersangkutan

f) *Functionality*, apakah system sudah sesuai dengan kebutuhan

g) *Reliability*, menunjukkan seberapa besar sistem dapat diandalkan untuk melakukan suatu proses yang dapat dipercaya

h) *Importance*, kepentingan informasi untuk digunakan

2.2.2 Informasi

2.2.2.1 Pengertian Informasi

Menurut Jugiyanto definisi informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya[2]. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data-item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu.

Menurut Davis dalam Abdul Kadir informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau saat mendatan[6].

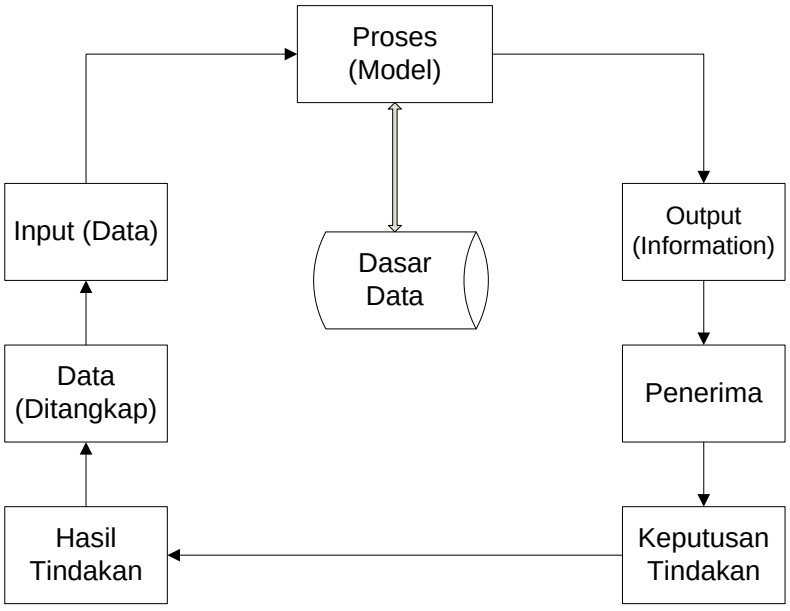
Menurut Tata Sutabri informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan[7].

Dari definisi informasi diatas maka diambil kesimpulan bahwa informasi adalah data yang diolah menjadi sebuah bentuk yang lebih berguna bagi penerima dan dapat bermanfaat untuk mengambil keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata.Informasi diperoleh setelah data-data mentah diproses atau diolah.

2.2.2.2 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lainnya yang akan membuat sejumlah data kembali.

Data tersebut akan ditanggap sebagai input, diproses kembali lewat suatu model dan membentuk suatu siklus. Menurut John Burch disebut dengan siklus informasi (*information cycle*) atau siklus ini disebut juga dengan siklus pengolahan data (*data processing cycles*).

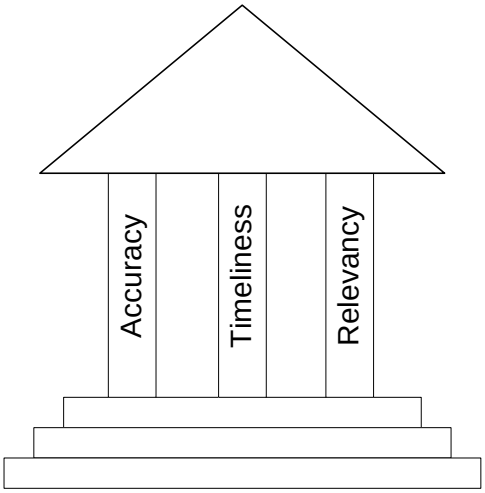


Gambar 2. 4 *Siklus Informasi [2]*

2.2.2.3 Kualitas Informasi

Menurut Jogiyanto kualitas dari suatu informasi (*Quality of Information*) tergantung dari tiga hal, yaitu informasi harus akurat (*Accurate*), tepat pada waktunya (*Timeliness*), dan relevan (*Relevance*)[2].

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti Informasi yang datang pada penerima tidak boleh terlambat, informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa ini mahalnya nilai informasi disebabkan harus cepatnya informasi tersebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.
3. Relevan, berarti Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda, misalnya informasi mengenai sebabmusabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi relevan unuk akuntan.



Gambar 2. 5 *Kualitas Informasi [2]*

Menurut Guimares [15], yang menjadi faktor atau komponen dalam mengukur sebuah kualitas informasi adalah:

a) *Availability,* apakah sistem dapat diakses pada saat dan dimana ia diperlukan.

b) *Usability,* informasi memiliki makna yang berguna untuk membantu user dalam menyelesaikan pekerjaannya.

c) *Understandability*, informasi yang ditampung dalam laporan adalah kemudahannya untuk segera dipahami oleh pemakai.

d) Relevance, informasi akan relevan jika memberikan manfaat bagi user

e) *Format*, informasi yang dihasilkan dalam bentuk atau susunan yang mudah dipahami/dimengerti.

f) *Conciseness*, tingkat penggabungan atau pengikhtisaran informasi (ringkas dan padat).

g) *Accuracy*, keterandalan dan ketepatan informasi.

h) *Timeliness*, informasi yang diterim tidak boleh terlambat, kekinian informasi.

i) *Completeness*, informasi yang dihasilkan atau dibutuhkan memiliki kelengkapan yang baik. Karena bila informasi yang dihasilkan sebagian-sebagian, tentunya akan mempengaruhi dalam pengambilan keputusan atau menentukan tindakan secara keseluruhan, sehingga akan berpengaruh terhadap kemampuan untuk mengontrol atau memecahkan suatu masalah dengan baik

j) *Consistency*, perkiraan dari *response time* dan kemampuan dalam proses mengalirkan data.

2.2.2.4 Nilai Informasi

Menurut Jogiyanto nilai dari informasi (*Value of Information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya[2]. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan di dalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya, karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak di dalam perusahaan. Pengukuran nilai informasi biasanya dihubungkan dengan analisis cost effectiveness atau cost benefit.

Menurut Sutabri nilai informasi ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkan informasi tersebut[7]. Suatu informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Sebagian besar informasi tidak dapat ditaksir secara pasti nilai keuntungannya (dalam satuan uang), tetapi kita dapat menaksir nilai efektifitas dari informasi tersebut".

Menurut Hemingway ada dua faktor yang menentukan nilai informasi yaitu kemampuan untuk menghasilkan dan mengunakan informasi dan keuntungan dari penggunaan informasi itu sendiri[8].

2.2.2.5 Sistem Informasi

Menurut Jogiyanto sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasasi, dan menyediakan pihak luar tertentu dengan laporan - laporan yang diperlukan[2].

Menurut George M.Scott sistem informasi adalah sistem yang diciptakan oleh para analisis dan manajer guna melaksanakan tugas khusus tertentu yang sangat esensial bagi berfungsinya organisasi[9].

Jadi dapat disimpulkan bahwa sistem informasi adalah suatu alat yang membantu dalam menyediakan informasi bagi penerimanya dan untuk membantu dalam pengambilan keputusan bagi manajemen didalam operasi perusahaan sehari-hari dan informasi yang layak untuk pihak luar perusahaan.

Secara teori, penerapan sebuah Sistem Informasi memang tidak harus menggunakan komputer dalam kegiatannya. Tetapi pada prakteknya tidak mungkin sistem informasi yang sangat kompleks itu dapat berjalan dengan baik jika tanpa adanya komputer. Sistem Informasi merupakan sistem pembangkit informasi. Dengan integrasi yang dimiliki antar subsistemnya, sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya.

2.2.3 Basis Data

Menurut Connolly dan Begg basis data merupakan sekumpulan data yang saling terhubung secara logis serta pengertian dari data tersebut yang dirancang untuk keperluan sebuah perusahaan[10]. Basis data merupakan tempat penyimpanan data yang besar yang digunakan secara terus menerus oleh banyak pengguna. Basis data digunakan sebagai pengganti dari penggunan file yang tidak saling terhubung, yang menyebabkan redudansi data. Semua data diintegrasikan untuk mengurangi duplikasi.

Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

2.2.3.1 Data, Informasi, dan Basis Data

Data merupakan bahan baku yang belum mempunyai makna kepada pengguna sehingga perlu diolah untuk dapat menghasilkan sesuatu yang bermakna. Menurut Jogiyanto yang dimaksud dengan data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata[2].

Kenneth C.Louden da Jane P.Louden menyatakan bahwah data adalah sekumpulan fakta yang mewakili diolah kedalam format yang bisa dimengerti dan digunakan[11].

Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data[12].

Dari pengertian diatas dapat diambil kesimpulan bahwa data merupakan bentuk yang masih mentah sehingga masih perlu diolah lebih lanjut. Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (*database*).

Sedangkan menurut George Tsu-der Chou basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang[13].

Menurut Anthoni J. Fabbri dan A. Robert Schwab basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data[14].

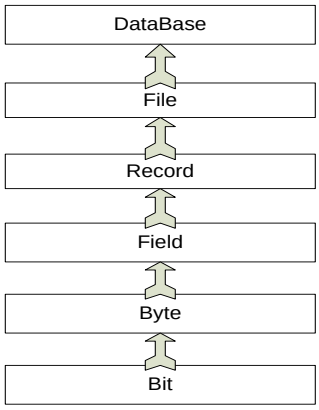
Menurut Ramez Elmasri mendefinisikan basis data lebih dibatasi pada arti implisit yang khusus, yaitu:

a. Basis data merupakan penyajian suatu aspek dari dunia nyata (*real world*).
b. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
c. Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa user dan beberapa aplikasi yang sesuai dengan kepentingan user.

Dari beberapa definisi-definisi tersebut, dapat dikatakan bahwa basis data memounyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa user untuk berbagai kepentingan.

2.2.3.2 Hirarki Data

Data harus disusun secara teratur agar pengolahannya dapat dilakukan dengan baik dan efisien. Pengorganisasian data dapat dibagi dalam enam tingkatan, yaitu :



Gambar 2. 6 *Hirarki Data [6]*

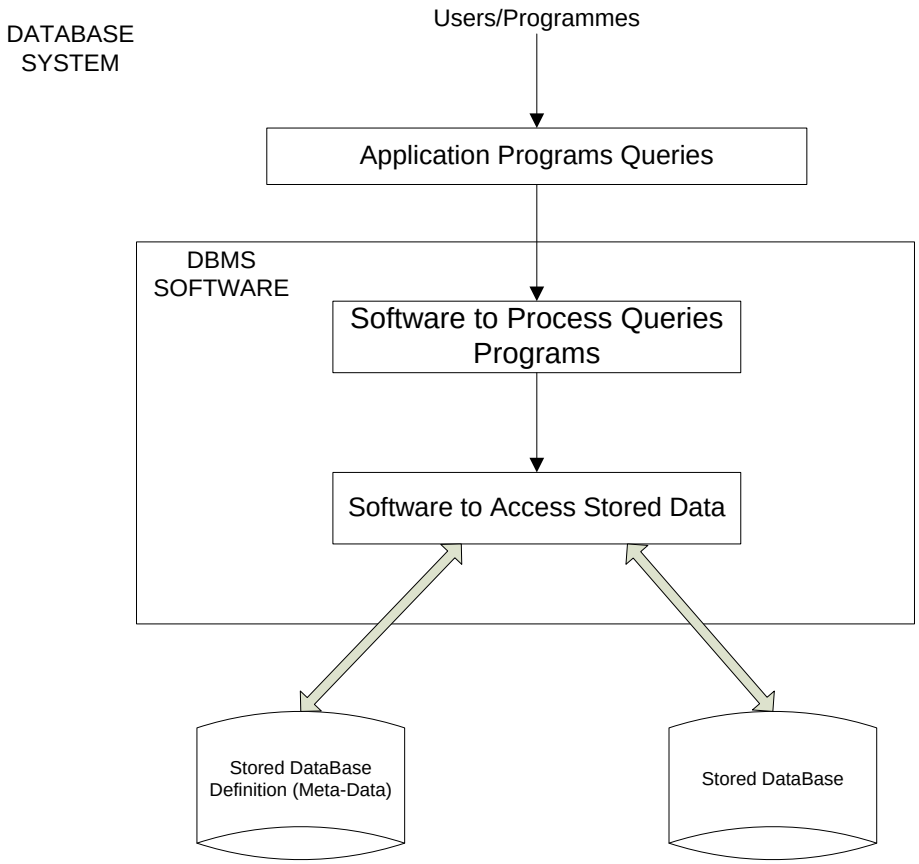
Dari gambar 2.6 dapat dijelaskan sebagai berikut :

a) Database yaitu kumpulan dari beberapa file/ tabel yang saling berhubungan antara file yang satu dengan file yang lain.
b) File yaitu kumpulan dari record yang saling berkaitan dan memiliki format field yang sama dan sejenis.
c) Record yaitu kumpulan dari field yang menggambarkan suatu unit data indifidu tertentu.
d) Field yaitu suatu atribut dari record yang menunjukan suatu item dari data nilai record sebuah field.
e) Byte yaitu atribut dari field yang berupa karakter yang membentuk nilai dari sebuah field.
f) Bit yaitu bagian terkecil dari data secara keseluruhan yaitu huruf karakter Ascil (American Standard Code From Information Interchange) nol atau satu yang merupakan komponen pembentuk byte.

2.2.3.3 Sistem Basis Data

Menurut Jogiyanto Sistem Basis Data adalah Suatu sistem informasi yang mengintegarasikan kumpulan dari data yang berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi[2]. Komponen-komponen utama sebuah sistem basis data :

a. Perangkat Keras (Hardware).
b. Sistem Operasi (Operating Sistem).
c. Basis Data (Data Base).
d. Sistem Pengolahan Basis Data.
e. Pemakai (User).
f. Aplikasi (perangkat lunak) lain (bersifat opsional).



Gambar 2. 7 *Konsep Sistem Basis Data [17]*

Keuntungan penggunaan Sitem Basis Data / DBMS :

a. Data *Independence*: Program Aplikasi tidak harus, idealnya, akan terkena rincian representasi data dan penyimpanan, DBMS memberikan pandangan abstrak dari data yang menyembunyikan rincian tersebut.

b. Efisien Akses Data: Sebuah DBMS menggunakan berbagai teknik canggih untuk menyimpan dan mengambil data secara efisien. Fitur ini sangat penting jika data yang disimpan pada perangkat penyimpanan eksternal.

c. Integritas Data dan Keamanan: Jika data selalu diakses melalui DBMS, DBMS dapat menegakkan batasan integritas. Misalnya, sebelum memasukkan informasi

gaji karyawan, DBMS dapat memeriksa bahwa anggaran departemen tidak terlampaui. Selain itu, dapat menegakkan contmls akses yang mengatur data apa yang dapat dilihat oleh kelas yang berbeda dari pengguna.

d. Data Administrasi: Ketika beberapa pengguna berbagi data, sentralisasi administrasi data dapat menawarkan perbaikan signifikan. Profesional berpengalaman yang memahami sifat dari data yang dikelola, dan bagaimana berbagai kelompok pengguna menggunakannya, dapat bertanggung jawab untuk mengatur representasi data untuk meminimalkan redundansi dan untuk fine-tuning penyimpanan data untuk membuat pengambilan efisien.

e. Concurrent Akses dan Crash Recovery: Sebuah jadwal DBMS akses bersamaan dengan data sedemikian rupa sehingga pengguna bisa memikirkan data yang sedang diakses oleh hanya satu pengguna pada satu waktu. Selanjutnya, DBMS melindungi pengguna dari efek kegagalan sistem.

f. Aplikasi Waktu Pengembangan Reduced: Jelas, DBMS mendukung fungsi penting yang umum untuk banyak aplikasi mengakses data dalam DBMS. Hal ini, dalam hubungannya dengan antarmuka tingkat tinggi untuk data, memfasilitasi pengembangan aplikasi cepat. Aplikasi DBMS juga cenderung lebih kuat daripada sejenis aplikasi yang berdiri sendiri karena tugas-tugas penting yang ditangani oleh DBMS (dan tidak harus *debugged* dan diuji dalam aplikasi).

2.2.3.4 Pemodelan Data

Pemodelan sistem memainkan peranan yang penting dalam pengembangan sistem. Pemodelan data kadang-kadang disebut pemodelan *database* karena model data kadang-kadang diimplementasikan sebagai sebuah database. Pemodelan data dapat di gambarkan dengan ERD (Entity Relationship Diagram).

Menurut Whitten, *Entity Relationship Diagram* (ERD) merupakan pemodelan data yang menggunakan beberapa notasi untuk menggambarkan data yang berhubungan dengan entity dan relationship[18].

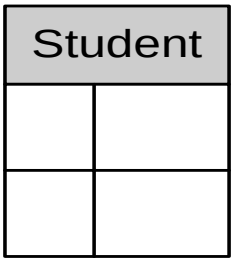
Ada beberapa konsep dasar bagi setiap model data :

a) Entitas *(Entity)*

Entity adalah sebuah class dari orang, tempat, objek, kejadian, atau konsep mengenai apa yang dilakukan untuk mengambil dan menyimpan data. *Entity* dilambangkan dengan sebuah lambang persegi.

Beberapa kategori dari entity adalah sebagai berikut:

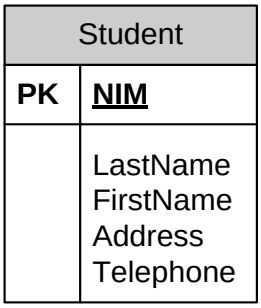
1. Orang : Vendor, pelanggan, karyawan, dan lain-lain.
2. Tempat : Ruangan, kantor cabang, gedung, dan lain-lain.
3. Objek : Produk, mesin, bahan dasar, dan lain-lain.
4. Kejadian: Invoice, pemesanan, penjualan, dan lain-lain.
5. Konsep : Stok, dana, account, kualifikasi, dan lain-lain.



Gambar 2. 8 *Contoh Entity dalam ERD*

b) Atribut (*Attribute*)

Attribute adalah sebuah properti dari sebuah *entity* atau relasi (*relationship*). Setiap *attribute* memiliki tipe data yang mendefinisikan nilai dari data tersebut.



Gambar 2. 9 *Conoh Atribut dalam ERD*

c) Indentifikasi (*Identification*)

Dengan banyaknya *instance* yang dimiliki oleh suatu enity maka diperlukan suatu *key* yang unik untuk mengindetifikasikan setiap *instance* berdasarkan data dari atribut. Yang dimaksud *key* adalah suatu atribut atau sekumpulan atribut yang mengasumsikan nilai yang unik dari setiap bagian dari *entity* dan seringkali disebut *identifier*.

Candidate key adakah suatu *key* yang memiliki kemungkinan untuk dijadikan *primary key. Primary key* adalah *candidate key* yang unik dan mengidentifikasikan sebuah bagian dari *entity. Alternate key* adalah *candidate key* yang tidak dijadikan *primary key.*

d) Relasi (*Relationship*)

Relationship adalah sebuah hubungan antara satu atau lebih *entity. Cardinality* adalah jumlah minimum dan maksimum dari keberadaan suatu *entity* yang mungkin direlasikan dengan *entity* lain. *Degree* adalah sejumlah *entity* yang berpartisipasi dalam sebuah *relationship. Foreign key* adalah sebuah *primary key* yang digunakan oleh *entity* lain untuk mengidentifikasikan instansi dari sebuah *relationship.*

Menurut Whitten di dalam penggambaran ERD, terdapat *cardinality*, yaitu suatu angka minimum atau maksimum dari kejadian suatu *entity* yang mungkin berhubungan dengan kejadian *entity* yang lainnya. Karena semua *relationship* adalah dua arah, *cardinality* harus didefinisikan dalam dua arah untuk setiap *relationship.*

Non-identifying *relationship* adalah hubungan dimana setiap *entity* partisipan mempunyai *primary key* independen-nya masing-masing. *Relationship* tipe ini

digambarkan dengan garis putus-putus dan biasa disebut *strong entity*. Sementara indentifying *relationship adalah* hubungan dimana setiap enity partisipan mempunyai *primary key* yang sama. *Relationship* tipe inin digambarkan dengan garis yang tidak terputus-putus dna biasa disebut *weak entity*.

e) Generalisasi (*Generalization*)

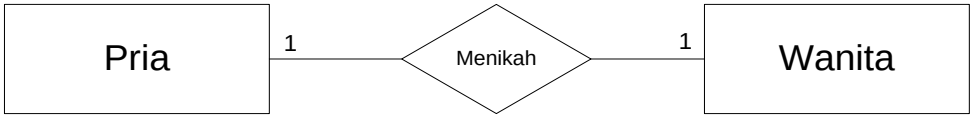
Generalisasi atau generalization ialah sebuah konsep dimana atribut-atribut yang umum bagi beberapa tipe *entity* dikelompokkan kedalam *entity* mereka masing-masing.

Beberapa macam hubungan antar relasi, antara lain:

a. Satu ke Satu (*One to One*)

Bentuk relasi antara satu entitas dengan sejumlah satu ke entitas dengan jumlah yang sama.

Contoh :



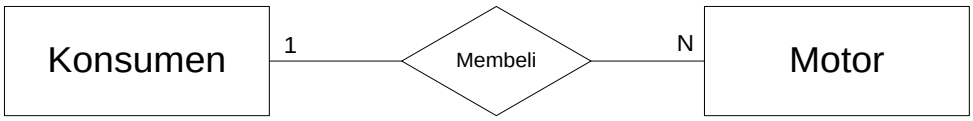
Gambar 2. 10 *Contoh relasi satu ke satu*

Keterangan : Pria dan wanita dengan relasi menikah. Satu pria menikah hanya dengan satu wanita. Satu wanita menikah hanya dengan satu pria. Diasumsikan tidak diperbolehkan adanya poligami.

b. Satu ke Banyak (*One to Many*)

Bentuk relasi dari entitas dengan jumlah satu ke entitas lain yang berjumlah lebih dari satu (Entitas dengan banyak alternatif tujuan).

Contoh :



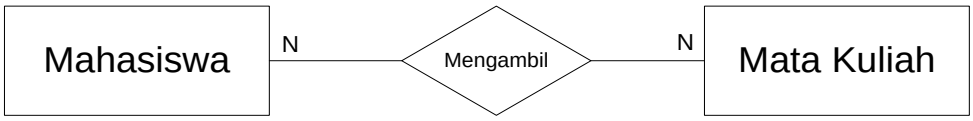
Gambar 2. 11 *Contoh relasi satu ke banyak*

Keterangan : Konsumen dan motor dengan relasi membeli. Satu konsumen dapat membeli banyak motor. Satu motor hanya dapat dibeli oleh satu konsumen.

c. Banyak ke Banyak (*Many to Many*)

Bentuk relasi yang mendeskripsikan permasalahan yang komplek yaitu hubungan antara entitas yang berjumlah lebih dari satu dengan entitas yang sama.

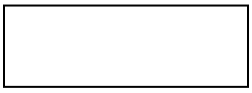
Contoh :

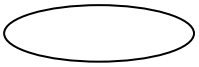
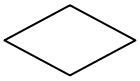


Gambar 2. 12 *Contoh relasi banyak ke banyak*

Keterangan : Mahasiswa dan mata kuliah dengan relasi mengambil. Satu mahasiswa dapat mengambil banyak mata kuliah. Satu mata kuliah dapat diambil oleh banyak mahasiswa.

Tabel 2. 1 *Tabel Notasi yang digunakan pada Entity Relational Diagram (ERD)*

No	Nama Simbol	Fungsi	Simbol
1	Persegi panjang	Menyatakan suatu entitas	

2	Elips	Menyatakan atribut	
3	Belah ketupat	Menyatakan relasi	
4	Garis	Menyatakan hubungan antar entitas dengan relasi atau hubungan antar entitas dengan relasi	

2.2.3.5 Analisis Sistem

Analisis sistem didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

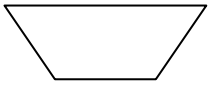
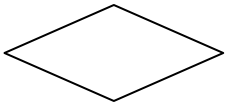
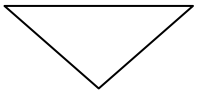
Sedangkan menurut Abdul Kadir analisis sistem merupakan tahapan yang dimulai karena adanya permintaan terhadap sistem baru[6]. Dimana, permintaan dapat datang dari seorang manajer di luar departemen sistem informasi atau dari pihak eksekutif yang melihat adanya masalah atau menemukan adanya peluang baru. Sehingga tujuan utama analisis sistem adalah untuk menetukan hal-hal detail yang akan dikerjakan oleh sistem yang akan diusulkan.

Pada model analisis terdapat perangkat lunak yang dapat digambarkan dalam bentuk sebagai berikut:

1. *Flowmap*

Flowmap adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Flowmap berguna untuk membantu analis dan programer untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif pengoperasian. Biasanya flowmap mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Tabel 2. 2 *Tabel Notasi yang digunakan pada Flowmap*

No	Nama Simbol	Fungsi	Simbol
1	Document	Menyatakan suatau formulir/document	
2	Trapesium	Menyatakan proses yang dilakukan secara manual	
3	Persegi panjang	Menyatakan proses yang dilakukan secara otomatis	
4	Belah ketupat	Menyatakan proses pengambilan keputusan benar atau salah	
5	Segitiga	Menyatakan arsip	
6	Garis penghubung	Menggambarkan aliran document	

2. Konsep Dasar Berorientasi Obyek

Sebelumnya mari kita definisikan dulu pengertian obyek. Obyek adalah “benda” secara fisik atau konseptual, yang dapat kita temui di sekeliling kita. Obyek adalah riil. Contoh obyek adalah orang, hardware, software, dokumen dan lain-lain.

Setiap obyek mempunyai dua ciri, yaitu atribut (property atau data) yang menjadi ciri khas dari suatu obyek (what they have) dan method (behavior/function), yaitu apa yang dapat dilakukan oleh obyek (what they do).

Berorientasi Obyek (object oriented) berarti permasalahan didefinisikan melalui istilah dari obyek yang mengkapsulasi data (atribut) dan perilaku (behavior), yaitu melalui paradigma/pendekatan obyek.

Selain object, ada beberapa istilah yang akan membantu untuk memahami pengertian kita dalam skripsi ini:

a. Class, yaitu kumpulan obyek yang sejenis. Secara lebih lugas obyek adalah instant dari sebuah class, atau dengan pengertian lain dengan class kita menggambarkan property dan behavior dari tipe obyek.

b. Inheritance, adalah penurunan atribut atau method dari suatu obyek class ke obyek class lainnya.

c. Polymorphisme, berasal dari bahasa Yunani yang berarti banyak bentuk.

Dalam konsep ini memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu obyek untuk melakukan suatu aksi atau tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda. Secara sederhana bisa juga disebut : satu interface, banyak aksi.

Metodologi adalah cara sistematis untuk mengerjakan pekerjaan analisis dan desain. Metodologi berorientasi obyek adalah metode penyelesaian masalah dengan menggunakan pendekatan berorientasi obyek.

Metodologi berorientasi obyek pertama kali muncul pada pertengahan tahun 1970 dan terus berkelanjutan dikembangkan sampai saat ini. Pada tahun 1994 ada 72 lebih metode object oriented. Dengan berkembang pesatnya metode ini maka masyarakat object oriented menyadari perlunya standarisasi.

3. *Unified Modelling Language (UML)*

Pada Oktober 1994 Dr. James Rumbaugh yang mengembangkan Object Modelling Technique (OMT) bergabung dengan perusahaan Rational Software. Sebelumnya juga bergabung Grady Booch yang mengembangkan Object Modelling Design (OOD). Duet mereka pada Oktober 1995 menghasilkan Unified Method versi

0.8, yang menjadi cikal bakal dari UML (Unified Modelling language) sebagai bahasa pemodelan standar untuk aplikasi object oriented.

Pada tahun 2002 lahir UML versi 2.0 dengan penambahan dan penggantian diagram menjadi 13 buah diagram. Diagram-diagram ini terbagi menjadi 3 kategori :

a. Structural diagrams : menggambarkan elemen dari spesifikasi yang mengabaikan waktu. Terdiri dari : Class Diagram, Object Diagram, Component Diagram, Deployment Diagram, Composite Structure Diagram dan Package Diagram.

b. Behavior diagram : menggambarkan ciri-ciri behavior/method/function dari sebuah system atau business process. Terdiri dari : Use Case Diagram, Activity Diagram dan State Machine Diagram.

c. Interaction diagram : bagian dari behavior diagram yang menggambarkan object interactions. Terdiri dari : Communication Diagram, Interaction Overview Diagram, Sequence Diagram dan Timing Diagram.

Karena UML sangat fleksibel, ada juga cara melihat diagram UML berdasar kategori berikut :

a. Static Diagram : menunjukkan segi static dari system. Kategori ini sama dengan structural diagram.

b. Dynamic Diagram : menunjukkan bagaimana system berkembang setiap waktu. Meliputi state-machine diagram dan timing diagram.

c. Functional Diagram : menunjukkan detail dari perilaku (behavior) dan algoritma bagaimana system memenuhi perilaku yang diinginkannya. Kategori ini termasuk use case, interaction dan activity diagram.

3. Analisa dan perancangan berorientasi obyek

Analisa dan desain berorientasi obyek berarti merumuskan dan menyelesaikan masalah serta menghasilkan suatu hipotesa dan diagnosa (solusi), memodelkannya

dengan pendekatan/paradigma obyek (obyek adalah riil punya atribut/data dan perilaku).

Dalam melakukan analisa dan perancangan sistem berorientasi obyek penulis menggunakan UML (Unified Modelling Language) untuk memodelkannya. Sedangkan alat (tool) visual modelling yang digunakan untuk menggambarkan model analisa dan perancangan adalah Microsoft Visio 2007. Implementasi perangkat lunak menggunakan bahasa pemrograman PHP.

a. Analisa Berorientasi Obyek (Object Oriented Analysis)

Object oriented analysis adalah metode analisis yang memeriksa requirements (syarat atau keperluan yang harus dipenuhi suatu sistem)

(Suhendar dan Hariman, 2002:11)

Dalam tahap ini kegiatan-kegiatan yang dilakukan dalam menganalisa sistem sebagai berikut :

- Menganalisa sistem yang ada dan mempelajari apa yang dikerjakan oleh sistem yang ada.
- Menspesifikasikan sistem yaitu spesifikasi masukan yang digunakan database yang ada, proses yang dilakukan dan keluaran yang dihasilkan.

Tujuan dari analisa berorientasi obyek yaitu untuk menentukan kebutuhan pemakai secara akurat.

Pendekatan-pendekatan yang dipakai dalam analisa berorientasi obyek antara lain :

- Pendekatan top down, yaitu memecahkan masalah ke dalam bagian-bagian terkecil atau per level sehingga mudah untuk diselesaikan.
- Pendekatan modul, yaitu membagi sistem ke dalam modul-modul yang dapat beroperasi tanpa ketergantungan.

- Penggunaan alat-alat bantu dalam bentuk grafik dan teks sehingga mudah untuk dimengerti serta dikoreksi apabila terjadi perubahan.

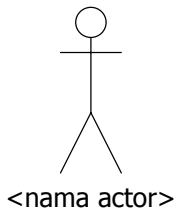
Pendekatan dalam analisa berorientasi obyek dilengkapi dengan alat-alat dan teknik-teknik yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang terdefinisi dengan baik dan jelas.

b. Use Case Diagram

Use case diagram menggambarkan kebutuhan sistem dari sudut pandang user. Digunakan untuk menggambarkan hubungan antara internal sistem dan eksternal sistem atau hubungan antara use case dan aktor.

1) Actor

Actor adalah sesuatu (entitas) yang berhubungan dengan sistem dan berpartisipasi dalam use case. Actor menggambarkan orang, sistem atau entitas eksternal yang secara khusus membangkitkan sistem dengan input atau masukan kejadian-kejadian, atau menerima sesuatu dari sistem. Actor dilukiskan dengan peran yang mereka mainkan dalam use case, seperti Staff, Kurir dan lain-lain.



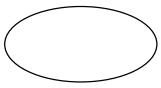
Gambar 2. 13 *Benuk Actor dalam UML*

Dalam use case diagram terdapat satu aktor pemulai atau initiator actor yang membangkitkan rangsangan awal terhadap sistem, dan mungkin sejumlah aktor lain yang berpartisipasi atau participating actor. Akan sangat berguna untuk mengetahui siapa aktor pemulai tersebut.

2) Use Case

Use case yang dibuat berdasar keperluan aktor merupakan gambaran dari "apa" yang dikerjakan oleh sistem, bukan "bagaimana" sistem mengerjakannya. Use case diberi nama yang menyatakan apa hal yang dicapai dari interaksinya dengan aktor.

Dalam UML use case dinotasikan dengan gambar :

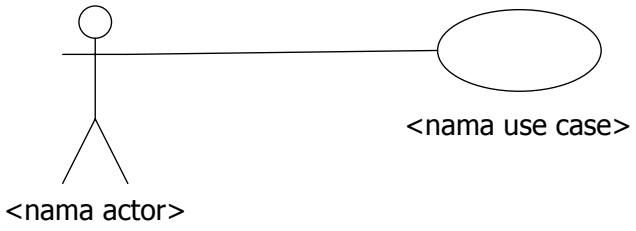


Gambar 2. 14 *Bentuk Use Case dalam UML*

3) Relationship

Relasi (relationship) digambarkan sebagai bentuk garis antara dua simbol dalam use case diagram. Relasi antara actor dan use case disebut juga dengan asosiasi (association). Asosiasi ini digunakan untuk menggambarkan bagaimana hubungan antara keduanya.

Relasi-relasi yang terjadi pada use case diagram bisa antara actor dengan use case atau use case dengan use case.



Gambar 2. 15 *Bentuk Relationship dalam UML*

Relasi antara use case dengan use case :

a. Include, pemanggilan use case oleh use case lain atau untuk menggambarkan suatu use case termasuk di dalam use case lain

(diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan <<include>>.

b. Extend, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak kontrol form dan mendeklarasikan ekstension pada use case utama. Atau dengan kata lain adalah perluasan dari use case lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis berpanah dengan tulisan <<extend>>.

c. Generalization/Inheritance, dibuat ketika ada sebuah kejadian yang lain sendiri atau perlakuan khusus dan merupakan pola berhubungan base-parent use case. Digambarkan dengan garis berpanah tertutup dari base use case ke parent use case.

c. *Activity Diagram*

Diagram aktivitas menggambarkan proses bisnis dan urutan aktivitas-aktivitas yang mendukung penggambaran tindakan sistem baik yang bersifat kondisional maupun paralel. Tindakan kondisional dilukiskan dengan cabang (branch) dan penyatuan (merge).

Sebuah branch memiliki sebuah transition masuk atau yang disebut dengan incoming transition dan beberapa transition keluar atau yang disebut dengan outgoing transition dari branch yang berupa keputusan-keputusan. Hanya satu dari outgoing transition yang dapat diambil, maka keputusan-keputusan tersebut harus bersifat mutually exclusive. [else] digunakan sebagai keterangan singkat yang menunjukkan bahwa transition “else” tersebut harus digunakan jika semua keputusan yang ada pada branch salah.

Sebuah merge memiliki banyak input transition dan sebuah output. Merge menandakan akhir dari suatu kondisi yang diawali dengan sebuah branch. Selain branch dan merge, di dalam diagram aktivitas terdapat pula fork dan join. Fork memiliki satu incoming transition dan beberapa outgoing transition. Sedangkan pada

join, outgoing transition diambil atau digunakan hanya ketika semua state pada incoming transition telah menyelesaikan aktivitasnya.

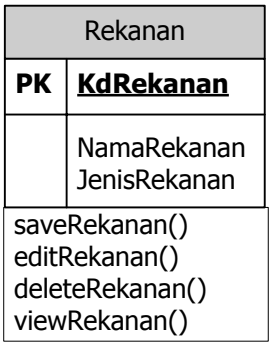
d. *Sequence Diagram*

Diagram yang menggambarkan bagaimana obyek berinteraksi dengan obyek lainnya melalui pesan (message) yang disampaikan, disusun dalam urutan kejadian atau waktu dan secara khusus berasosiasi dengan use case.

e. *Class Diagram*

Class diagram merupakan bagian yang paling penting dalm analisa dan perancangan berorientasi obyek. Dalam UML diagram kelas digunakan untuk memodelkan static structure dari sistem informasi.

Kelas merupakan himpunan dari obyek yang sejenis yang mempunyai atribut (attribute) dan perilaku (behaviors/method) yang sama. Atribut adalah sebuah nilai data karakteristik yang dimiliki oleh obyek sebuah kelas sedangkan method adalah perilaku atau operasi yang dikenakan oleh suatu kelas. Pada gambar kelas terdapat tiga bagiannya.



Gambar 2. 16 *Bentuk Class dalam UML*

Diagram kelas menggambarkan struktur obyek sistem, dimana diperlihatkan hubungan amtar mereka. Diagram kelas merupakan fondasi untuk component diagram dan deployment diagram.

Secara garis besar terdapat 3 jenis class. Ketiga jenis class tersebut dikelompokkan berdasarkan fungsi dan karakternya masing-masing, yaitu :

a. Entity Class Diagram

Merupakan paket utama dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk sistem dan menjadi landasan untuk menyusun basis data pada model data konseptual.



Gambar 2. 17 *Bentuk Entity Class dalam UML*

b. Control Class Diagram

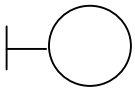
Berisi kumpulan kelas yang menjadi kontrol program termasuk koneksi dengan basis data dan merupakan kelas perantara atau penghubung antara entity class dengan kelas antar muka pemakai (interface).



Gambar 2. 18 *Bentuk Control Class dalam UML*

c. Boundary Class Diagram

Berisi kumpulan kelas yang menjadi interface antara pemakai (user) dengan sistem, seperti tampilan form untuk pencetakan.



Gambar 2. 19 *Bentuk Boundary Class dalam UML*

f. *Package Diagram*

Package (paket) adalah mekanisme pengelompokan yang digunakan untuk menandakan pengelompokan elemen-elemen model. Sebuah package dapat mengandung beberapa paket lain di dalamnya. Package digunakan untuk memudahkan pengorganisasian elemen-elemen model.

g. Perancangan Berorientasi Obyek (Object Oriented Design)

Object oriented design adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi obyek-obyek sistem atau subsistem (Suhendar dan hariman, 2001:11).

Perancangan berorientasi obyek merupakan proses spesifikasi yang terperinci atau pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk. Untuk mengembangkan suatu sistem baru digunakan dengan menguraikan hubungan proses-proses dalam bentuk diagram-diagram.

Perancangan berorientasi obyek bertujuan untuk :

a. Sistematika proses pendesainan
b. Menghasilkan pendesainan model program
c. Memberikan gambaran pemecahan masukan dengan efektif

2.2.3.6 Kamus Data

Kamus Data menurut Jogiyanto adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi[2]. Kamus data merupakan katalog fakta data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat berdasarkan arus data yang ada pada diagram alir data. Kamus data harus juga dapat mencerminkan keterangan yang jelas tentang data yang dicatat. Kamus data atau *data dictionary* disebut juga dengan istilah sistem *data dictionary.*

Adapun isi dari kamus data adalah:

a) Nama arus data

Kamus data dibuat berdasarkan arus data dalam DAD sehingga nama arus data harus dicatat di kamus data

b) Alias

Merupakan nama lain dari data yang harus dituliskan. Karena sering terdapat data yang sama tetapi mempunyai nama yang berbeda departemen yang satu dengan yang lainnya.

c) Bentuk data

Bentuk data berupa dokumen dasar laporan cetak, parameter dan field-field.

d) Arus Data

Menjelaskan dari mana arus data mengalir dan kemana data akan menuju.

e) Penjelasan

Untuk menjelaskan makna dari arus data yang dicatat pada kamus.

f) Periode

Menunjukan kapan terjadi arus data sampai dengan kapan proses program beserta laporan-laporan dapat dihasilkan.

g) Volume

Volume yang dicatat berupa volume rata-rata dan volume puncak suatu arus data.

h) Struktur data

Menunjukan arus data yang dicatat pada kamus data yang terdiri dari elemen-elemen data. Dalam kamus data dikenal dengan sebutan notasi. Notasi

merupakan kamus data yang mempunyai suatu bentuk untuk mempersingkat arti dari simbol yang dijelaskan.

2.2.3.7 Pengolahan Data

Menurut Moekijat pengolahan data adalah kegiatan pikiran dengan bantuan tangan atau dengan suatu peralatan dan mengikuti serangkaian langkah, perumusan atau pola tertentu untuk mengubah data sehingga data tersebut baik bentuk, susunan, sifat, atau isinya menjadi lebih berguna[19].

Menurut Mahyuzir pengolahan data adalah Suatu kesatuan prosedur-prosedur yang terlibat langsung dengan pengolahan data, prosedur- prosedur tersebut seperti prosedur penerimaan, pencatatan, penyimpanan, persiapan dan penyajian informasi yang di perlukan untuk mecapai tujuan, beliau juga menuturkan bahwa pengolahan data adalah segala macam manipulasiterhadap data agar data itu dapat berguna sesuai dengan yang di inginkan[20].

Menurut Budi Sutedjo pengolahan data adalah tahap ini merupakan tahap dimana data diolah sesuai dengan prosedur yang telahdimasukan[16].

Pengolahan Data adalah manifulasi dari data kedalam bentuk yang lebih berguna berarti. Dengan demikian dapat disimpulkan bahwa Pengolahan Data merupakan kegiatan yang dilakukan dengan menggunakan masukan berupa data dan menghasilkan informasi yang bermanfaat untuk tujuan sesuai dengan yang direncanakan.

Jadi, Pengolahan data adalah manipulasi data agar menjadi bentuk yanglebih berguna, pengolahan data ini tidak hanya melibatkan perhitungan numerik tetapi juga operasi - operasi, klasifikasi data dan perpindahan data suatu tempatketempat lain atau segala macam pengolahan terhadap data agar data itu bergunasesuai dengan diinginkan

2.2.4 Kuesioner

Angket atau kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pertanyaan tertulis kepada responden untuk dijawabnya. Kuesioner merupakan teknik pengumpulan data yang efisien. Kuesioner atau daftar pertanyaan disusun dengan memperhatikan atau menerapkan *Skala Likert,* yaitu skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang variabel penelitian. Dengan Skala Likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijalankan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan (Sugiyono, 2004).

Jawaban setiap item instrumen yang menggunakan *Skala Likert* mempunyai gradasi dari sangat positif sampai negatif, kemudian data diolah dengan *Skala Likert* dengan jawaban atas pertanyaan dengan skala nilai 1-5. Nilai yang dimaksud adalah skor atas jawaban responden, dimana skor yang digunakan adalah dengan kategori sebagai berikut.

Tabel 2. 3 *Kategori Skoring Variabel*

No	Kategori Skor	Skor
1	Sangat Setuju (SS)	5
2	Setuju (S)	4
3	Kurang Setuju (KS)	3
4	Tidak Setuju (TS)	2
5	Sangat Tidak Setuju (STS)	1

Ciri khas dari *Skala Likert* adalah bahwa semakin tinggi skor yang diperoleh seorang responden merupakan indikasi bahwa responden tersebut sikapnya makin positif terhadap objek yang diteliti.

Contoh Kasus :

Seorang mahasiswa gizi melakukan uji organoleptik sebuah produk dengan menggunakan Skala Likert. Aspek yang ingin diukur ialah cita rasa dari sebuah produk yang ia hasilkan, dari 70 panelis yang ia gunakan, berikut rangkuman hasil penilaian panelis.

1. Panelis yang menjawab sangat suka (5) = 2 orang
2. Panelis yang menjawab suka (4) = 8 orang
3. Panelis yang menjawab kurang setuju (3) = 15 orang
4. Panelis yang menjawab tidak suka (2) = 25 orang
5. Panelis yang menjawab sangat tidak suka (1) = 20 orang

Cara perhitungan skor tersebut di atas adalah sebagai berikut :

1. Panelis yang menjawab sangat suka (5) = 2 x 5 = 10
2. Panelis yang menjawab suka (4) = 8 x 4 = 32
3. Panelis yang menjawab kurang setuju (3) = 15 x 3 = 45
4. Panelis yang menjawab tidak suka (2) = 25 x 2 = 50
5. Panelis yang menjawab sangat tidak suka (1) = 20 x 1 = 20

Total skor = 157

Interpretasi skor perhitungan :

Sebagai tambahan informasi, bahwa jumlah skor tertinggi untuk item SANGAT SUKA ialah 5 x 70 = 350, sedangkan item SANGAT TIDAK SUKA ialah 1 x 70 = 70. Jadi, jika total skor penilaian panelis di peroleh angka 157, maka penilaian panelis terhadap cita rasa produk tersebut adalah : (157/350) x 100% = 44.86%, atau bisa dikategorikan sebagai cukup. Berikut kriteria interpretasi skor :

Angka 0% – 20% = Sangat lemah

Angka 21% – 40% = Lemah

Angka 41% – 60% = Cukup

Angka 61% – 80% = Kuat

Angka 81% – 100% = Sangat kuat

2.2.5 Perangkat Lunak yang Digunakan

2.2.5.1 Delphi

Delphi adalah suatu program berbasis bahasa Pascal yang berjalan dalam lingkungan Windows. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman Object Oriented Programming (OOP).



Gambar 2. 20 *Borland Delphi 7*

Langkah awal dari belajar Delphi adalah mengenal IDE (Integrated Developement Environment) Delphi yang merupakan Lingkungan Pengembangan Terpadu bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE programmer dapat membuat user interface, melakukan koding,

melakukan testing dan debuging serta menkompilasi program menjadi executable. Penguasaan yang baik akan IDE akan sangat membantu programmer dalam mengefektifkan tugastugasnya sehingga dapat bekerja dengan efisien.

IDE Delphi menggunakan model MDI (Multiple Document Interface). Berikut ini adalah gambar yang menunjukan bagian-bagian dan nama-nama jendela yang dapat tampil pada IDE Delphi. Sebagai langkah awal dari proses belajar, tidak semua jendela akan kita gunakan, tetapi hanya beberapa yang penting, sedangkan yang lainnya bersifat khusus. IDE atau lingkungan pengembangan terpadu keja Delphi7 terbagi menjadi 8 bagian utama, yaitu :

a) Main Window

Main Window atau jendela utama adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi 7 terbagi menjadi tiga bagian, yaitu : Main Menu, Toolbar dan Component Palette. Dengan menggunakan fasilitas dari menu utama (main menu) dapat memanggil atau menyimpan program. Pada dasarnya semua perintah yang diberikan dapat ditemukan pada bagian menu utama ini.

b) ToolBar

Delphi 7 memiliki beberapa toolbar yang masing-masing memiliki perbedaan fungsi dan setiap tombol pada bagian toolbar berfungsi sebagai pengganti suatu menu perintah yang sering digunakan. Pada kondisi default Delphi 7 memiliki tujuh bagian toolbar, antara lain: Standard, View, Debug, Custom, Componen Palette, Desktops dan Internet. Tombol-tombol yang terletak pada bagian toolbar dapat ditambah atau dikurangi sesuai kebutuhan.

c) Component Palette

Component Palette berisi kumpulan icon yang melambangkan komponen-komponen yang terdapat pada VCL (Visual Component Library). Pada Component Palette Anda

akan menemukan beberapa page control seperti Standard, Additional, Win32, System, Data Access dan lain-lain. Icon component terdapat di setiap page control. Page control ini dipakai untuk memilih icon component.

d) Form Designer

Form merupakan suatu objek yang dapat dipakai sebagai tempat untuk merancang dialog program aplikasi. Form berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari Component Palette. Pada saat Anda memulai Delphi 7, Delphi 7 akan memberikan sebuah form kosong yang disebut Form1. Sebuah form mengandung unit yang berfungsi untuk mengendalikan form dan Anda dapat mengendalikan komponen-komponen yang terletak dalam form dengan menggunakan Object Inspector dan Code Editor.

e) Object TreeView

Object TreeView menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam form, data module, atau frame. Object TreeView juga menampilkan hubungan logika antar komponen. Apabila Anda mengklik kanan salah satu item yang terdapat di dalam diagram pohon, Anda dapat melihat konteks menu komponen versi sebelumnya. Untuk mengakses menu secara penuh, klik kanan pada komponen yang sama dalam form, data module, atau frame.

f) Object Inspector

Object Inspector terdiri dari dua tab, yaitu Properties dan Events. Pada awalnya yang kelihatan hanya tab Properties sedangkan tab Events tidak kelihatan, untuk menampilkan klik tab Events. Object Inspector digunakan untuk mengubah properti atau karakteristik dari sebuah komponen pada posisi tab Properties, sedangkan pada posisi tab Events digunakan untuk memilih dan membuka events procedure.

g) Code Editor

Pada saat Anda memulai Delphi 7, code editor ini tidak kelihatan. Untuk menampilkannya klik icon Toggle pada toolbar sehingga code editor ditampilkan. Code Editor merupakan tempat di mana Anda dapat menuliskan kode program. Pada bagian ini Anda dapat menuliskan pernyataan-pernyataan dalam Object Pascal. Satu diantara keuntungan bagi pengguna Delphi 7 adalah bahwa Anda tidak perlu menuliskan semua kode-kode program sumber, karena Delphi 7 telah menyediakan kerangka penulisan sebuah program.

h) Code Explorer

Code Explorer merupakan fasilitas baru yang terdapat di dalam Delphi 7 yang tidak ditemukan pada versi-versi sebelumnya. Code Explorer digunakan untuk memudahkan pemakai berpindah antar file unit yang terdapat di dalam jendela Code Editor. Untuk menutup Code Explorer, klik tanda cross yang terdapat di sudut kanan atas, dan untuk membukanya kembali pilih menu View → Code Explorer dari menu utama atau klik kanan dalam jendela Code Editor kemudian pilih View Explorer.

2.2.5.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu system database (DBMS)

dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os Server, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source*

MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. *Command and Functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

7. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenskripsi.

8. *Scalability and Limits*

MySQL mampu menangani database skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unit soket (UNIX), atau Named Pipes (NT).

10. *Localisation*

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. *Clients and Tools*

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur Tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

14. *Create*

Pembuatan tabel dibuat dengan menggunakan perintah CREATE TABLE Format Pembuatan Tabel adalah :

```
CREATE TABLE nama_tabel(

kolom1 tipedata(ukuran) Option_Kolom,

kolom2 tipedata(ukuran) Option_Kolom,

...,

kolomN tipedata(ukuran)

) ENGINE=NamaEngine;
```

Option_Kolom tidak wajib disertakan. Option_Kolom yang diperbolehkan diantaranya : null/not null, primary key, auto_increment, unique, default dan comment. Contoh NamaEngine yaitu MyISAM dan InnoDB

15. *Read*

SELECT digunakan untuk mengambil data dari database. Bentuk umum pemanggilan SELECT adalah :

```
SELECT  * | {[DISTINCT] column|expression [alias], …}

FROM nama_table

[WHERE kondisi];
```

Keterangan :

a. * digunakan untuk menampilkan semua kolom yang ada dalam table
b. Daftar_kolom digunakan untuk mengampikan kolom tertentu saja
c. WHERE digunakan jika ingin membatasi data yang ditampilkan

16. *Update*

UPDATE digunakan untuk mengganti isi data. Bentuk umum penggunaan update adalah :

UPDATE nama_tabel

SET kolom1={isi|DEFAULT} [, kolom2={expr2|DEFAULT}] ...]

[WHERE kondisi_where]

[ORDER BY nama_kolom_pengurutan]

[LIMIT banyak_baris]

Keterangan :

a. WHERE digunakan untuk membatasi banyaknya baris yang diupdate
b. ORDER BY digunakan untuk melakukan pengurutan data yang akan diupdate.
c. LIMIT digunakan untuk membatasi baris yang akan diupdate (setelah data dibatasi dengan WHERE).

17. *Delete*

Penghapusan tabel dibuat dengan menggunakan perintah DELETE. Format penghapusan Tabel adalah :

DELETE From nama_table

[WHERE kondisi];

Where digunakan untuk membatasi banyaknya data yang akan dihapus.