BAB 2
TINJAUAN PUSTAKA

2.1 PT. Sinergi Kreasi Solusindo

PT. Sinergi Kreasi Solusindo adalah perusahaan yang bergerak dibidang Informasi Teknologi (IT) dan merupakan sebuah perusahaan jasa pembuatan website, program aplikasi dan sistem informasi. Perusahaan ini beralamat di Jalan Gunung Sahari Komplek Marinatama No. B15 Mangga Dua Jakarta Utara. PT. Sinergi Kreasi Solusindo merupakan perusahaan baru yang didirikan sejak awal tahun 2013.

2.1.1 Visi dan Misi

Visi dan misi yang terdapat pada PT. Sinergi Kreasi Solusindo adalah ;

1) Visi

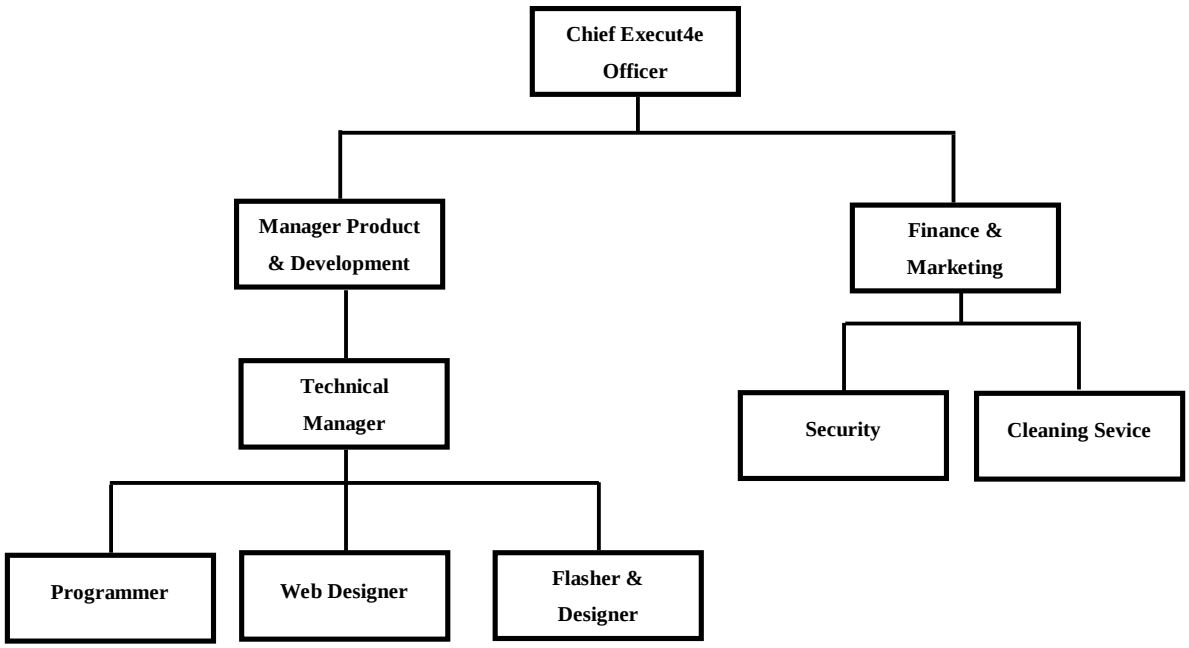
Menjadi perusahaan penyedia jasa teknologi informasi dan komunikasi terbaik di indonesia.

2) Misi

1. Mengembangkan solusi teknologi informasi yang berkesinambungan.
2. Mitra kerja yang terpercaya dan dapat diandalkan.
3. Memberikan *benefit* dan *value* bagi pelanggan dan seluruh *stakeholder*.
4. Membangun jaringan kerjasama untuk menumbuhkan industri teknologi informasi di Indonesia.

2.1.2 Struktur Organisasi

Adapun struktur organisasi pada PT. Sinergi Kreasi Solusindo dapat dilihat pada gambar 2.1.



Gambar 1.1 Struktur Organisasi

2.1.3 Deskripsi Jabatan

Deskripsi jabatan adalah penjabaran alur kerja dan tanggung jawab pegawai dalam menjalankan pekerjaannya sesuai dengan jabatannya. Dibawah ini adalah deskripsi jabatan yang ada di PT. Sinergi Kreasi Solusindo sebagai berikut :

1. *Chief Executive Officer* (CEO) merupakan jabatan tertinggi di suatu perusahaan dan mempunyai tugas untuk memimpin perusahaan dan bertanggung jawab untuk kestabilan perusahaan.
2. *Manager Product & Development* merupakan pihak yang bertanggung jawab akan semua yang terjadi dalam perusahaan dan pengelolaan operasional perusahaan.
3. *Finance & Marketing* bertugas dalam pengelolaan fungsi keuangan di perusahaan untuk membantu perusahaan dalam proses pengambilan keputusan yang mendukung pencapaian target *financial* perusahaan.

   Pengelolaan proses penjualan dan pemasaran untuk mencapai target penjualan dan mengembangkan pasar secara efektif dan efisien.

4. *Technical Manager* bertugas dalam pengelolaan teknisi perusahaan seperti mengatur dan mengawasi kelancaran proses bisinis perusahaan, membawahi *web designer, flasher and designer* serta *programmer*.
5. *Web Designer* bertugas mendesain web, mengkonversi dari aplikasi desain ke dalam HTML (Hypertext Markup Language).
6. *Flasher*& Designer bertugas mendesain animasi dalam pembuatan program.
7. *Programmer* bertugas menentukan strategi, sistem, kebijakan dan program yang akan diterapkan perusahaan.
8. *Security* bertugas melaksanakan pengamanan secara menyeluruh dilokasi perusahaan sesuai jadwal kegiatan.
9. *Cleaning Service* bertugas untuk pemeliharaan gedung bagian luar dan bagian dalam perusahaan sesuai jadwal kegiatan.

2.2 Landasan Teori

Pada landasan teori ini akan menerangkan mengenai teori-teori yang berhubungan dengan aplikasi sistem informasi periklanan di PT. Sinergi Kreasi Solusindo dengan memanfaatkan media jejaring sosial mengenai pengertian iklan serta teori-teori pendukung lainnya. Berikut ini akan dibahas mengenai teori-teori yang menjadi sumber kajian ilmu dari aplikasi sistem informasi periklanan.

2.2.1 Konsep Dasar Sistem

Istilah sistem bukanlah hal yang asing bagi kebanyakan orang. Seringkali sistem mengacu pada komputer seperti IBM PC atau Macintosh, tetapi juga bisa ke arah yang lebih luas seperti sistem tatasurya atau bahkan ke hal-hal yang lebih spesifik seperti sistem respirasi mamalia.

Pada dasarnya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan [6]. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

2.2.2 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu :

1. Tujuan

   Setiap sistem memiliki tujuan, entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem lain berbeda-beda.

2. Masukan (*Input*)

   Masukan sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak sejara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkah contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa dari pelanggan).

3. Proses

   Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa pemanasan bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

4. Keluaran (*Output*)

   Keluaran merupakan hasil dari pemrosesan.

5. Mekanisme Pengendalian dan Umpan Balik

   Mekanisme pengendalian diwujudkan dengan menggunakan umpan balikm yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan. Dalam bentuk yang sederhana, dilakukan perbandingan antara keluaran sistem dan keluaran yang dikehendaki (standar). Jika terdapat penyimpangan terletak pada proses, maka prosesnyalah yang diperbaiki.

6. Batas

   Yang disebut batas sistem adalah pemisah antara sistem dan daerah diluar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian oleh pelanggan, gerakan pesaing, dan ketersediaan dana dari bank. Tentu saja, batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbatasan dana.

7. Lingkungan

   Lingkungan adalah segala sesuatu yang berada di luar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem. Lingkungan bagi sebuah organisasi dapat berupa vendor, pelanggan, pemilik, pemerintah, bank, dan bahkan pesaing.

2.2.3 Pengertian Sistem Informasi

Sesungguhnya yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer-Based Information Sistems* atau CBIS). Dalam prakteknya, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang paling penting.

Ada beragam definisi informasi, sebagaimana tercantum pada tabel 2.1. dari berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja),

ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan [6].

Tabel 1.1 Pengertian Sistem Informasi

Sumber	Definisi
Alter (1992)	Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
Bodnar dan Hopwood (1993)	Sistem informasi adalah kumpulan perangkat keras dan oerangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
Gelinas, Oram, dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada pemakai.
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.
Turban, McLean, dan Weterbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
Wikinson (1992)	Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sararan perusahaan. (A.S. and Shalahuddin, 2011)

2.2.4 Komponen Sistem Informasi

Dalam suatu sistem informasi terdapat komponen-komponen seperti :

1. Perangkat keras (*hardware*)

   Mencakup peranti-peranti fisik seperti komputer dan printer.

2. Perangkat lunak (*software*) atau program

   Sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.

3. Prosedur

   Sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

4. Orang

   Semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.

5. Basis data (*database*)

   Sekumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.

6. Jaringan komputer dan komunikasi data

   Sistem penghubung yang memungkinkan sumber dipakai ecara bersama atau diakses oleh sejumlah pemakai.

Pada prakteknya, tidak semua sistem informasi mencakup keseluruhan komponen-komponen tersebut. Sebagai contoh, sistem informasi pribadi yang hanya melibatkan sebuah pemakai dan sebuah komputer tidak melibatkan fasilitas jaringan dan komunikasi. Namun, sistem informasi grup kerja yang melibatkan sejumlah orang dan sejumlah komputer, memerlukan sarana jaringan dan komunikasi.

2.2.5 Pengertian Iklan

Iklan merupakan sebuah proses komunikasi yang bertujuan untuk membujuk orang untuk mengambil tindakan yang menguntungkan bagi pihak pembuat iklan. Iklan ditujukan untuk mempengaruhi perasaan, pengetahuan, makna, kepercayaan, sikap, pendapat, pemikiran dan citra konsumen yang berkaitan dengan suatu produk atau merek, tujuan periklanan ini bermuara pada upaya untuk dapat mempengaruhi perilaku konsumen dalam membeli sebuah produk yang ditawarkan.

Kata Iklan sendiri berasal dari bahasa yunani, yang artinya adalah upaya menggiring orang pada gagasan. Adapun pengertian secara *komprehensif* atau luas adalah semua bentuk aktifitas untuk menghadirkan dan mempromosikan ide,

barang ataupun jasa secara *nonpersonal* melalui media yang dibayar oleh sponsor tertentu [2].

Menurut pakar periklanan dari Amerika, *S. William Pattis* iklan adalah setiap bentuk komunikasi yang dimaksudkan untuk memotivasi dan mempromosikan produk dan jasa kepada seseorang atau pembeli yang potensial. Tujuannya adalah mempengaruhi calon konsumen untuk berfikir dan bertindak sesuai dengan keinginan si pemasang iklan. Pengertian lainnya, iklan adalah seni menyampaikan apa yang ditawarkan atau dijual untuk mendapatkan perhatian dan menempatkan produk secara unik kedalam pikiran konsumen dengan alat bantu [3].

Pengertian antara iklan dan periklanan mempunyai persamaan dan perbedaan. Persamaannya adalah bahwa keduanya merupakan pesan yang ditujukan kepada khalayak. Perbedaannya yaitu iklan lebih cenderung kepada produk atau merupakan hasil dari periklanan, sedangkan periklanan merupakan keseluruhan proses yang meliputi penyiapan, perencanaan pelaksanaan, dan pengawasan penyampaian iklan.

Iklan merupakan bagian dari bauran promosi *(promotion mix)* sedangkan bauran promosi adalah bagian dari bauran pemasaran *(marketing mix)* dimana *marketing mix* meliputi *product*, *price*, *place*, *promotion.* Dalam bauran promosi sendiri dikenal ada lima unsur komunikasi yang utama [2] yaitu :

a. *Advertising*

   Semua bentuk penyajian pesan-pesan penjualan yang bersifat nonpersonal, berupa penjualan tentang barang atau jasa yang dibayar oleh suatu perusahaan.

b. *Sales Promotion*

   Berbagai bentuk insentif jangka pendek yang diberikan produsen untuk mendorong keinginan konsumen agar tertarik dan mencoba atau membeli suatu produk atau jasa yang ditawarkan.

c. *Public Relations*

   Berbagai macam program yang dibuat untuk memelihara, menciptakan, dan mengembangkan citra dan image perusahaan atau merek sebuah produk.

d. *Personal Selling*

Interaksi langsung dengan satu atau beberapa calon pembeli dengan melakukan presentasi, menjawab pertanyaan, dan menerima pesanan dari pembeli.

e. *Direct Marketing*

Interaksi langsung dengan konsumen melalui penggunaan surat, telepon, fax, *e-mail*, dan alat komunikasi nonpersonal lainnya untuk melakukan komunikasi secara langsung agar mendapat tanggapan dari calon konsumen.

Periklanan merupakan bagian untuk memenuhi fungsi pemasaran. Untuk dapat menjalankan fungsi pemasaran, maka periklanan tentu saja tidak sekedar memberikan informasi kepada khalayak tapi juga ditujukan untuk mempengaruhi perasaan, pengetahuan, makna, kepercayaan, sikap dan citra konsumen yang berkaitan dengan suatu produk atau merek. Citra produk yang terbaik biasanya tercipta melalui kegiatan-kegiatan periklanan. Dan juga segmen pasar yang dipilih sangat menentukan corak periklanan yang akan diluncurkan atau media yang harus dipilih.

2.2.5.1 Elemen Periklanan

Dalam periklanan, terdapat 5 pemain yang memegang peranan penting:

a. *The Advertiser*

Adalah orang yang ingin menyampaikan pesan kepada pasar. Pengiklan mempunyai tugas untuk mementukan target *audience*-nya, memilih media yang akan digunakan, menentukan *budget*-nya dan menentukan lama-nya waktu kampanye produk-nya. Pengiklan juga yang akan mempekerjakan biro iklan untuk mengkomunikasikan pesan yang ingin disampaikan kepada target audience.

b. *Advertising Agency*

Pengiklan menyewa sebuah biro iklan karena biro iklan biasanya lebih mengerti mengenai strategi dan konsep kreatif, pengetahuan mengenai media dan lain sebagainya sehingga pesan yang akan disampaikan pengiklan dapat disampaikan dengan baik ke target audience.

Biro iklan sendiri terdiri dari beberapa tipe yaitu :

1. *Independent Creative Services*

   Bentuk usaha periklanan perorangan atau organisasi yang memfokuskan diri pada pelayanan jasa kreatif sampai ke produksi. Dibagi atas *Freelance* (*fotografer, copywriter, desainer*) dan Biro Kreatif (bukan perorangan tetapi sebuah team kerja).

2. *Medium Advertising Agency*

   Sebuah usaha periklanan yang berskala menengah dan mempunyai jasa pelayanan yang lengkap namun pengiklan boleh memilih pelayanan yang mereka inginkan saja. Misalnya pengiklan hanya membutuhkan jasa pembuatan website atau brosur. Bekerja berdasarkan project.

3. *Full Services Advertising Agency*

   Usaha periklanan yang berskala besar yang memberikan pelayanan secara lengkap bekerja berdasarkan kontrak bukan per-*project*. Biasanya sebuah biro iklan dikontrak untuk satu masa kampanye.

4. *Relodex Advertising Agency*

   Bentuk usaha periklanan yang *flexible* yang tidak memiliki staff yang tetap. Biasanya terdiri dari beberapa profesional yang sudah memiliki pekerjaan tetap di biro iklan lain.

5. *Media Broker*

   Usaha periklanan yang memfokuskan diri pada penyediaan ruang dan waktu pada media.

6. *In House Advertising Agency*

   Bentuknya hanya berupa diisi dalam sebuah perusahaan. Biasanya digunakan pada *department store* dan *mall* atau perusahaan yang mempunyai *event* berkala dalam jangka waktu yang dekat.

c. The Media

Adalah tempat atau waktu yang akan digunakan oleh biro iklan untuk menyampaikan pesannya ke target *audience*. Media yang akan digunakan (bisa

lebih dari satu) akan menyampaikan pesan secara konsisten sesuai dengan konsep kreatif biro iklan.

d. Vendors

Adalah sekelompok pendukung biro iklan. Mereka adalah *freelance, consultant* media dan para *designer professional* yang bekerja sendiri seperti *photographer, copywriters* dan lainnya.

e. The Target Audience

Adalah sasaran dari pengiklan dan biro iklan. Semua strategi dan konsep selalu dimulai dari konsumen atau target market.

2.2.5.2 Tujuan Periklanan

Aspek pertama yang paling penting sebelum merumuskan strategi periklanan adalah sebuah sasaran atau tujuan. Tujuan itu tergantung pada apa yang ingin dicapai oleh klien. Penetapan tujuan periklanan merupakan aspek penting yang mendapat perhatian serius. Tanpa tujuan yang baik, tidak mungkin mengarahkan dan mengendalikan keputusan dengan efektif dan efisien.

Tujuan periklanan berfungsi sebagai alat komunikasi dan koordinasi, memberikan kriteria dalam pengambilan keputusan serta sebagai alat evaluasi. Selain itu tujuan juga akan sangat membantu dalam komunikasi dan membuat suatu garis antara keputusan strategis dan taktis. Secara umum tujuan perusahaan mengiklankan produknya adalah dalam rangka [2] :

a. Menciptakan kesadaran pada suatu merek di dalam benak konsumen. *Brand awarness* yang tinggi merupakan kunci untuk mencapai *brand equity* yang kuat. Para pemasar harus menyadari bahwa tanpa *Brand awarness* yang tinggi akan sulit untuk mendapatkan pangsa pasar yang tinggi.
b. Mengkomunikasikan informasi kepada konsumen mengenai keunggulan suatu merek. Manfaat ini berhubungan dengan keunggulan dari sebuah produk dibanding produk lain.
c. Mengasosiasikan suatu merek dengan perasaan serta emosi tertentu. Tujuannya, agar ada hubungan emosi antara konsumen dengan suatu merek.

d. Membuat perilaku, artinya perilaku konsumen dapat dibentuk melalui kampanye periklanan.
e. Mengembangkan atau mengubah citra atau personalitas dari sebuah merek. Sebuah merek terkadang mengalami keterpurukan dimata konsumen sehingga perlu diperbaiki citra atau *image*-nya yang dilakukan melalui periklanan.
f. Mengembangkan persepsi positif calon konsumen yang diharapkan kelak dapat menjadi pembeli potensial.
g. Mengarahkan konsumen untuk membeli produk. Tetapi yang harus disadari adalah iklan bukan segalanya, karena keberhasilan suatu merek di pasar tidak hanya tergantung pada periklanannya tapi juga ditentukan oleh elemen pemasaran lainnya.

Dari sekian banyak tujuan perusahaan dalam periklanan seperti yang telah disebut di atas, hanya beberapa tujuan saja yang harus dipilih. Tujuan tersebut harus benar-benar sesuai dengan apa yang diinginkan perusahaan terhadap kampanye produknya, semakin efektif dan efisien tujuan maka kemungkinan kampanye periklanan berhasil akan semakin terbuka.

Sebagaimana tujuan manajemen, tujuan periklanan harus dapat dioperasionalkan. Tujuan iklan yang baik tidak saja efektif bagi penentuan kriteria untuk mengambil keputusan, tetapi juga memberikan standar untuk membandingkannya dengan hasil akhir dari sebuah kampanye.

2.2.5.3 Jenis-Jenis Iklan

Bermacam-macamnya tujuan dari sebuah iklan yang dibuat oleh sebuah perusahaan maka berdampak pada berbedanya jenis-jenis dari sebuah iklan. Iklan yang sering muncul diberbagai media dan umumnya dibuat oleh perusahaan periklanan adalah sebagai berikut :

a. Iklan Komersial

   Iklan komersial adalah iklan yang bertujuan untuk mendukung kampanye pemasaran suatu produk atau jasa. Iklan komersial ini sendiri terbagi menjadi beberapa macam, yaitu [4] :

1. Iklan Strategis

   Digunakan untuk membangun merek. Hal itu dilakukan dengan mengkomunikasikan nilai merek dan manfaat produk. Perhatian utama dalam jangka panjang adalah memposisikan merek serta membangun pangsa pikiran dan pangsa pasar. Iklan ini mengundang konsumen untuk menikmati hubungan dengan merek serta meyakinkan bahwa merek ini ada bagi para pengguna.

2. Iklan Taktis

   Memiliki tujuan yang mendesak. Iklan ini dirancang untuk mendorong konsumen agar segera melakukan kontak dengan merek tertentu. Pada umumnya iklan ini memberikan penawaran khusus jangka pendek yang memacu konsumen memberikan respon pada hari yang sama.

b. Iklan *Corporate*

   Iklan yang bertujuan membangun citra suatu perusahaan yang pada akhirnya diharapkan juga membangun citra positif produk-produk atau jasa yang diproduksi oleh perusahaan tersebut [4].

   Iklan *Corporate* akan efektif bila didukung oleh fakta yang kuat dan relevan dengan masyarakat, mempunyai nilai berita dan biasanya selalu dikaitkan dengan kegiatan yang berorientasi pada kepentingan masyarakat. Iklan *Corporate* merupakan bentuk lain dari iklan strategis ketika sebuah perusahaan melakukan kampanye untuk mengkomunikasikan nilai-nilai korporatnya kepada publik [4].

   Iklan *Corporate* sering kali berbicara tentang nilai-nilai warisan perusahaan, komitmen perusahaan kepada pengawasan mutu, peluncuran merek dagang atau logo perusahaan yang baru atau mengkomunikasikan kepedulian perusahaan terhadap lingkungan sekitar.

c. Iklan Layanan Masyarakat

   Iklan Layanan Masyarakat merupakan bagian dari kampanye sosial marketing yang bertujuan menjual gagasan atau ide untuk kepentingan atau pelayanan masyarakat [4]. Biasanya pesan Iklan Layanan Masyarakat berupa ajakan, pernyataan atau himbauan kepada masyarakat untuk melakukan atau

tidak melakukan suatu tindakan demi kepentingan umum atau merubah perilaku yang tidak baik supaya menjadi lebih baik, misalnya masalah kebersihan lingkungan, mendorong penghargaan terhadap perbedaan pendapat, keluarga berencana, dan sebagainya [4].

Iklan layanan masyarakat juga menyajikan pesan sosial yang bertujuan untuk membangkitkan kepedulian masyarakat terhadap sejumlah masalah yang harus mereka hadapi, yakni kondisi yang dapat mengancam keserasian dan kehidupan mereka secara umum [4]. Pesan tersebut dengan kata lain bermaksud memberikan gambaran tentang peristiwa dan kejadian yang akan berakibat pada suatu keadaan tertentu, baik yang bersifat positif maupun negatif.

2.2.5.4 Media Iklan

Dalam periklanan, hal yang paling penting adalah dimana iklan nantinya akan ditempatkan. Karena itu diperlukan yang namanya media. Media dalam periklanan sangatlah beragam. Akhir-akhir ini bahkan ditemui banyak sekali media dengan bentuk yang juga beragam.

a. Media Cetak

Merupakan kumpulan berbagai media informasi yang dibuat dan disampaikan kepada khalayak sasaran melalui bentuk-bentuk tulisan atau cetakan dan seringkali disertai gambar sehingga dapat dilihat maupun dibaca [5].

Media cetak selain memiliki kelebihan juga memiliki kelemahan. Kekuatan dan kelemahan media itu harus diketahui secara mendalam dan menjadi bahan pertimbangan tim kreatif karena akan sangat berkaitan erat dengan teknis eksekusi kreatifnya. Contoh media cetak yang utama adalah surat kabar, majalah, tabloid, brosur, pamplet dan sebagainya.

1. Kelebihan Majalah :
   a. Mampu menjangkau segmen pasar tertentu yang terspesialisasi.
   b. Mampu mengangkat produk-produk yang diiklankan sejajar dengan persepsi khalayak sasaran terhadap prestise majalah yang bersangkutan.
   c. Merupakan media yang memiliki usia edar yang paling panjang.

d. Majalah yang dimiliki khalayak biasanya disimpan hingga bertahun-tahun sebagai referensi.
e. Memiliki keunggulan untuk memuat naskah iklan yang agak panjang dan detail.
f. Format majalah memungkinkan pengembangan pekerjaan kreatif dengan memanfaatkan beberapa halaman sekaligus, sisipan dan desain-desain lainnya.
g. Kualitas visual majalah sangat prima karena umumnya dicetak di atas kertas berkualitas tinggi.
h. Media yang efektif untuk menyiarkan pesan iklan yang berbau promosi penjualan seperti kupon-kupon yang akan diundi, contoh produk, dan kartu-kartu petunjuk.

2. Kelebihan Koran :
   a. *Market Coverage*, dapat menjangkau daerah-daerah perkotaan sesuai dengan cakupan pasarnya (nasional, regional atau lokal).
   b. *Comparison Shopping (catalog value)*, menyangkut kebiasaan konsumen membawa surat kabar sebagai referensi untuk membeli barang sewaktu berbelanja.
   c. *Positive Consumer Attitudes*, konsumen umumnya memandang surat kabar memuat hal-hal aktual yang perlu segera diketahui khalayak pembacanya.
   d. *Flexibility*, pengiklan dapat bebas memilih pasar mana (dalam cakupan geografis) yang akan diprioritaskan.
   e. Dengan demikian dapat memilih media mana yang cocok. Usia iklan lebih panjang karena dapat dibaca berhari-hari. Kebiasaan orang menyimpan surat kabar dan dibaca oleh 3-8 orang.
   f. *Target audience- nya* lebih luas, karena masyarakat pada umumnya pasti membaca koran setiap harinya. Dari masyarakat kelas atas sampai masyarakat kelas bawah, baik laki-laki maupun perempuan, serta pambaca koran tidak dibatasi oleh umur (untuk segala umur).
   g. Mudah dibawa kemanapun, ringan dan praktis.

b. Media Televisi

Kelebihan media televisi antara lain :

1. Pemirsa dapat diseleksi menurut jenis program dan waktu tayang.
2. Jangkauan massal yang menimbulkan efisiensi biaya untuk menjangkau setiap kepala.
3. Media regional mampu menjangkau wilayah luas. Efektif untuk tes pasar dan peluncuran produk.
4. Disaksikan sambil beraktivitas.
5. Lebih menarik, berupa gambar bergerak dan mengandung suatu cerita.
6. Menimbulkan dampak yang kuat terhadap konsumen, dengan tekanan pada sekaligus dua indera, penglihatan dan pendengaran.
7. Mampu menciptakan kelenturan bagi pekerjaan-pekerjaan kreatif dengan mengkombinasikan gerakan, kecantikan, suara, warna, drama, dan humor.
8. Mempunyai kemampuan yang kuat untuk mempengaruhi persepsi khalayak sasaran**.**
9. Jutaan orang menonton televisi secara teratur.
10. Masyarakat meluangkan waktunya di muka televisi sebagai sumber berita, hiburan dan sarana pendidikan.
11. Calon pembeli lebih “percaya” pada perusahaan yang mengiklankan produknya di televisi daripada yang tidak sama sekali. Ini adalah cerminan bonafiditas perusahaan.

c. Papan Reklame

Kelebihan dan efektifitas :

1. Cakupan geografisnya luas.
2. Mampu melokalisasikan atau mengkonsentrasikan wilayah tertentu.
3. Repetisinya tinggi : orang lalu lalang, atau iklannya lewat di depan orang orang untuk iklan transportasi.
4. Dapat diletakkan di dekat penyalurnya.
5. Relatif rendah biayanya.

6. Jumlah khalayak secara keseluruhan banyak.
7. Kontinuitas : kesinambungan media menyampaikan pesan iklan sesuai dengan tuntutan strategi periklanan. Billboard memiliki kesinambungan yang baik mengingat lokasinya yang tetap, dan di Indonesia umumnya masa kontrak berakhir untuk jangka waktu satu tahun.
8. Kemampuan media mengulang pesan yang sama terhadap khalayak sasaran saat mulai dilupakan.
9. Papan reklame mepunyai kemampuan untuk tampil secara mencolok dan tiba-tiba. Dengan ukuran yang besar, papan reklame mampu meyakinkan khalayak sasaran bahwa produknya benar-benar baik karena diiklankan secara serius, mahal dan *bonafide*.
10. Mampu menampilkan produk persis seperti aslinya. Karena ukurannya yang besar, media ini mampu menciptakan *smash impact* yang kuat sekali, sangat berguna untuk pengenalan produk baru.

d. Media Internet

Dengan semakin berkembangnya teknologi internet, semakin banyak pula alternatif cara sebuah iklan atau informasi ditampilkan dan didapatkan. Sekarang, iklan tidak hanya dalam bentuk konvensional yang menggunakan media kertas seperti koran, majalah, tabloid atau buku. Sekarang, iklan dapat dicari melalui media elektronik misalnya radio, televisi, dan *website* iklan di internet.

Setiap media memiliki kelebihan dan kekurangan. Misalnya, iklan melalui radio dan televisi. Iklan yang disiarkan cukup *up to date* karena dapat didengar atau dilihat oleh konsumen, namun apabila terlewatkan tidak akan dapat didengar atau dilihat kembali. Berbeda dengan koran, majalah atau media cetak yang lain. Iklan yang dikandungnya mungkin agak terlambat karena iklan hari sebelumnya baru akan tercetak di edisi berikutnya, namun pembaca dapat menyimpan iklan tersebut dalam waktu yang dapat ditentukan dengan menyimpan koran atau majalah tersebut.

Media internet berusaha mewakili dan meminimalkan kelemahan-kelemahan yang ada. Iklan yang tampilkan berusaha selalu *up to date*, yaitu setiap ada iklan

yang di *upload* akan seketika itu dapat dibaca oleh pengunjung. Data atau iklan hari ini tidak akan hilang walau sudah ditambah dengan iklan berikutnya karena semua data tersimpan dalam database. Data yang ada dalam database ini masih dapat diakses kembali oleh pengunjung menggunakan fasilitas pencarian atau arsip data yang disedaiakan dalam *website* tersebut.

2.2.6 Jejaring Sosial

Jejaring sosial adalah struktur sosial yang terdiri dari elemen-elemen individual atau organisasi. Jejaring ini menunjukan jalan dimana mereka berhubungan karena kesamaan sosialitas, mulai dari mereka yang dikenal sehari-hari sampai dengan keluarga**.**

Jejaring sosial merupakan sebuah bentuk layanan internet yang ditujukan sebagai komunitas online bagi orang yang memiliki kesamaan aktivitas, ketertarikan pada bidang tertentu, atau kesamaan latar balakang tertentu. Jejaring sosial lazim disebut sebagai jaringan pertemanan.

Layanan *social network* biasanya berbasis web, dilengkapi dengan beragam fitur bagi penggunanya agar dapat saling berkomunikasi dan berinteraksi. Contoh situs *social networking* yang populer antara lain *facebook* dan *twitter*.

2.2.6.1 Facebook

Facebook merupakan sebuah situs web jejaring sosial populer yang diluncurkan pada 4 Februari 2004. *Facebook* didirikan oleh Mark Zuckerberg, seorang mahasiswa Harvard kelahiran 14 Mei 1984 dan merupakan mantan murid Ardsley High School. Sejarah *facebook* berawal ketika Mark Zuckerberg membuat situs jejaring sosial facebook yang berisikan foto-foto kelulusan teman-teman di kampusnya. Pada mulanya pengunaannya hanya diperuntukkan bagi mahasiswa dari Harvard College.

Dalam dua bulan selanjutnya, keanggotaannya diperluas ke sekolah lain di wilayah Boston (Boston College, Universitas Boston, MIT, Tufts), Rochester, Stanford, NYU, Northwestern, dan semua sekolah yang termasuk dalam 4y League. Banyak perguruan tinggi lain yang selanjutnya ditambahkan berturut-

turut dalam kurun waktu satu tahun setelah peluncurannya. Mahasiswa yang memiliki alamat *e-mail* suatu universitas (seperti: .edu, .ac, .uk, dll) dari seluruh dunia dapat juga bergabung dengan situs jejaring sosial ini.

Facebook saat ini memiliki lebih dari 350 juta pengguna di seluruh dunia yang mana sekitar 55 juta pengguna melakukan *update* status *Facebook* tiap harinya. *Facebook* juga merupakan jejaring sosial yang menyediakan layanan unggah foto ke penggunanya.

Lebih dari 2,5 juta foto yang diunggah oleh pengguna *Facebook* tiap harinya. Tercatat ada beberapa negara yang melarang kehadiran jejaring sosial Facebook di negaranya. Negara-negara tersebut adalah China, Korut, Pakistan, Tajikistan, Kongo, Syria, dan Vietnam.

2.2.6.2 Twitter

Twitter adalah suatu layanan *social networking* yang termasuk dalam kategori *microblogging* yang dioperasikan dan dimiliki oleh Twitter Inc. Hingga saat ini ada sekitar 100 juta pengguna *twitter* di seluruh dunia. *Twitter* di perkenalkan oleh Jack Dorsey dan rekan-rekannya yaitu Biz Stone, dan Evan Williams pada tahun 2006. *Twitter* sendiri berpusat di San Bruno, California yang dekat dengan San Francisco yaitu daerah dimana pertama kali situs *twitter* ini dibuat.

Twitter mulai melejit pada tahun 2007 setelah diadakan festival *South by Southwest* (SXSW). Karena pada saat *event* berlangsung, status dari pengguna *twitter* mengalami peningkatan yang sangat pesat dari 20 ribuan kicauan per hari menjadi 60 ribuan. Lebih dari 400.000 kicauan yang tampil per-kuatal pada tahun 2007 tersebut. Kemudian berkembang menjadi 100 juta kicauan per-kuartal pada 2008. Pada akhir tahun 2009 menjadi 2 milyar per kuartal kicauan sudah dikirim-tampil dan selanjutnya berkembang dan berkembang hingga saat ini.

2.2.7 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi

tersedia saat dibutuhkan [6]. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa *file* teks ataupun *Database Management Sistem* (DMBS). Kebutuhan basis data dalam sistem informasi meliputi :

1. Masukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

2.2.8 DBMS

DBMS (*Database Management Sistem*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data [6]. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani *backup* data

Karena pentingnya data bagi suatu organisasi/perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis menangani DBMS yang disebut sebagai DBA (*Database Administrator*).

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational DBMS* yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana reasional model dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan *open source* saat ini berbasis *Relational* DMBS atau RDBMS.

Pada tahun 1980-an mulai berkembang *Object Oriented* DBMS (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman

berorientasi objek. Secara umum dapat diartikan bahwa OODBMS merupakan DBMS yang memandang data sebagai suatu objek. Saat ini OODBMS juga cukup berkembang namun belum dapat menggeser kepopuleran RDBMS.

Berikut ini adalah 4 macam DBMS versi komersial yang paling banyak digunakan di dunia saat ini, yaitu :

1. Oracle
2. Microsoft SQL Server
3. IBM DB2
4. Microsoft Access

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut:

1. MySQL
2. PostgreSQL
3. Firebird
4. SQLite

Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS.

2.2.9 SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS [6]. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

SQL mulai berkembang pada tahun 1970-an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini :

Tabel 1.2 Perkembangan SQL

No	Tahun	Nama
1	1986	SQL-86
2	1989	SQL-89
3	1992	SQL-92

4	1999	SQL:1992
5	2003	SQL:2003
6	2006	SQL:2006
7	2008	SQL:2008

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

Berikut ini adalah contoh pengaksesan data pada DBMS dengan SQL yang secara umum terdiri dari 4 hal sebagai berikut:

1. Memasukkan data (*insert*)

```
INSERT INTO Tabel_mahasiswa
(nim, nama, tanggal_lahir)
VALUES
('10109227', 'Citra', '1992-10-10');
```

Query di atas digunakan untuk memasukkan data mahasiswa dengan NIM 10109227, nama Citra, dan tanggal lahir 10 Oktober 1992 ke tabel "Tabel_mahasiswa".

2. Mengubah data (*update*)

```
UPDATE Tabel_mahasiswa
SET
Tanggal_lahir='1993-10-10'
  WHERE
  Nim='10109227';
```

Query di atas digunakan untuk mengubah data tanggal lahir mahasiswa dengan NIM = 10109227 menjadi 10 Oktober 1993 dalam tabel "Tabel_mahasiswa".

3. Menghapus data (*delete*)

```
DELETE FROM Tabel_mahasiswa
  WHERE
  Nim='10109227';
```

Query di atas dugunakan untuk menghapus data mahasiswa dengan NIM =10109227 dari tabel “Tabel_mahasiswa”.

4. Menampilkan data (*select*)

```
SELECT nim,nama
FROM
Tabel_mahasiswa
  WHERE
  Nim=’10109227’;
```

Query di atas digunakan untuk menampilkan data mahasiswa yang terimpan dalam “Tabel_mahasiswa” dengan NIM = 10109227.

2.2.10 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [6]. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.
3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

2.2.11 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (*relationship*)dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara teknis kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

   Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

   Secara teknis, sebuah kelas saat program dieksekusi akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat *runtime* yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat *runtime* (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari *event*, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata. Contoh metode atau operasi misalnya *Read, Write, Move, Copy,* dan sebagainya.

4. Atribut (*attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

   Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.12 Pengenalan UML

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang

menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

Pada perkembangan teknil pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan digaram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Seperti yang kita ketahui di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan, lingkunan dan konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi, belum tentu pandangan yang perbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin digambarkan. Secara analogi jika dengan bahasa yang digunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalahhal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan interpretasi di dalam bidang sistem informasi merupakan hal yang sangat wajar.

2.2.13 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya.

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian disaat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shlaer dan Stephe Mellor (Shlaer-Mellor, 1998) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen,1991), dan masih banyak lagi.

Buku terkenal yang juga berkembang selanjutnya adalah karangan Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang fokus pada *use case* dan proses pengembangan. Sekitar lima tahun kemudian muncul buku yang membahas mengenai metodologi berorientasi objek yang diikuti dengan buku-buku yang lainnya. di dalamnya juga membahas mengenai konsep, definisi, notasi, terminologi, dan proses mengenai metodologi berorientasi objek.

Karena banyak metodologi-metodologi yang berkembang pesat saat itu, maka munculah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modelling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran 4ar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML).

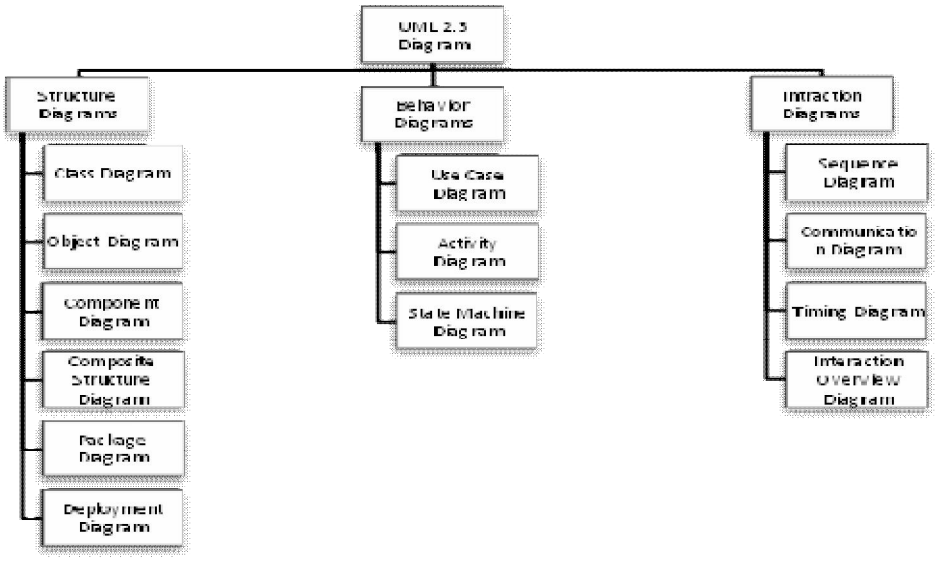
Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997

UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya.

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification,* UML *Infrastructure,* UML *Supersctructure,* dan *Object Constraint Language* (OCL). Seluruh spesifikasi tersebut dapat diakses di website http://www.omg.org.

2.2.14 Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar



Gambar 1.2 Diagram UML

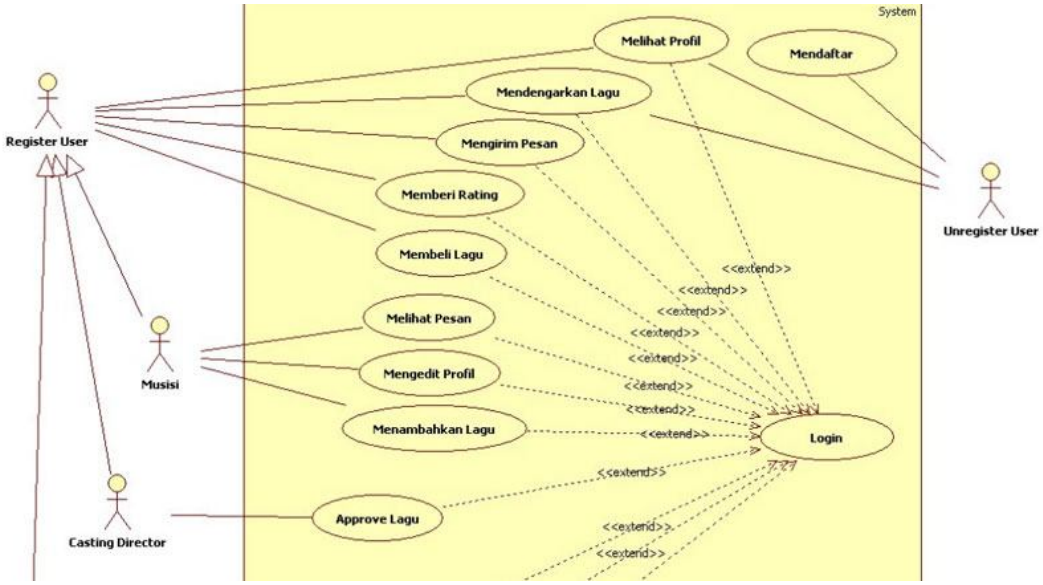
Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.14.1 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



Gambar 1.3 Use Case Diagram

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case.*

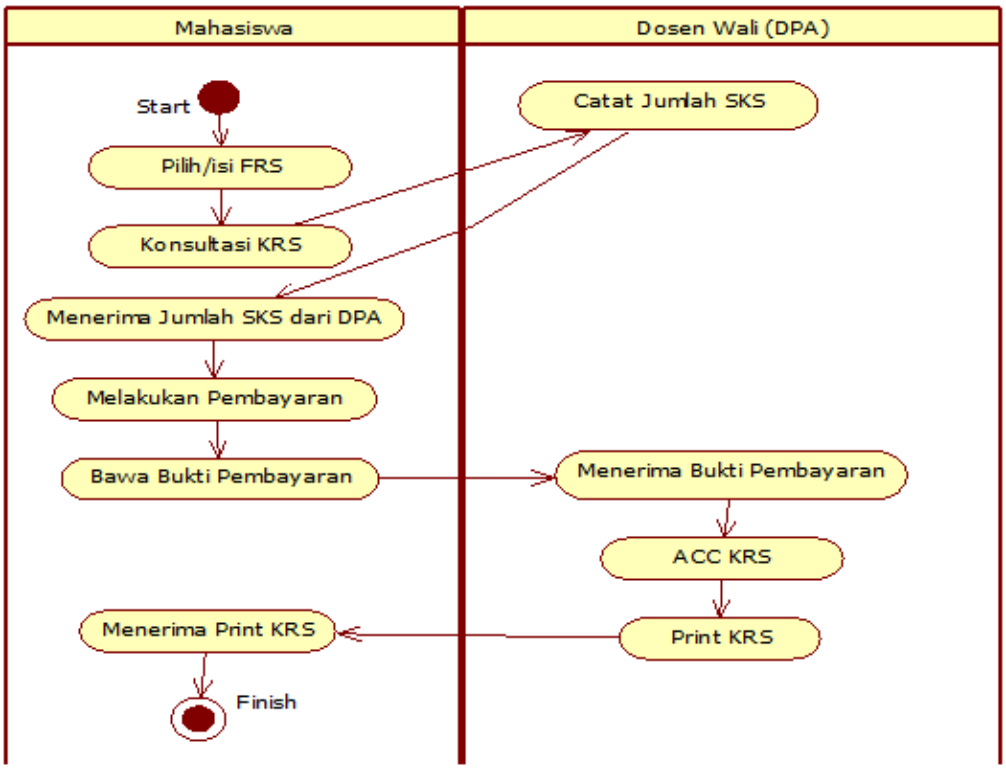
Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.2.14.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

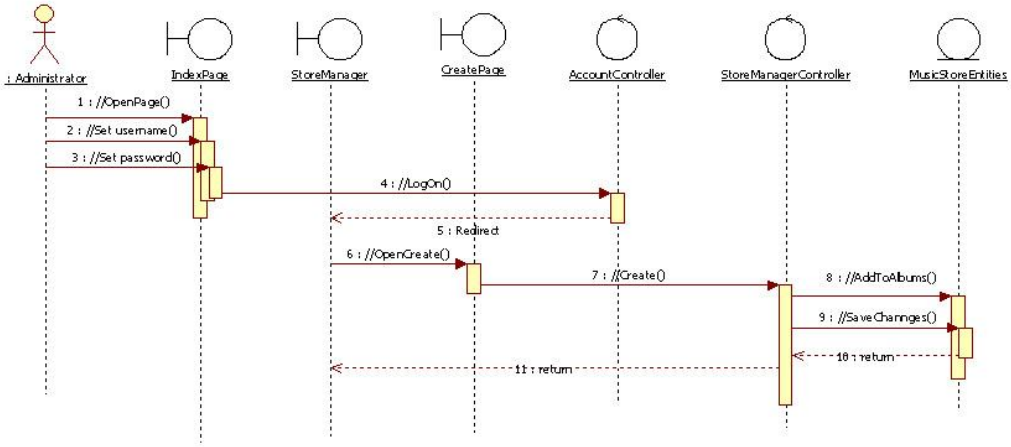


Gambar 1.4 Activity Diagram

2.2.14.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.



Gambar 1.5 Sequence Diagram

Penomoran pesan berdasarkan urutan iteraksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.2.14.4 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan

variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas ebaiknya memiliki jenis-jenis kelas berikut :

1) Kelas Main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2) Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3) Kelas yang diambil dari pendefinisian *usecase*

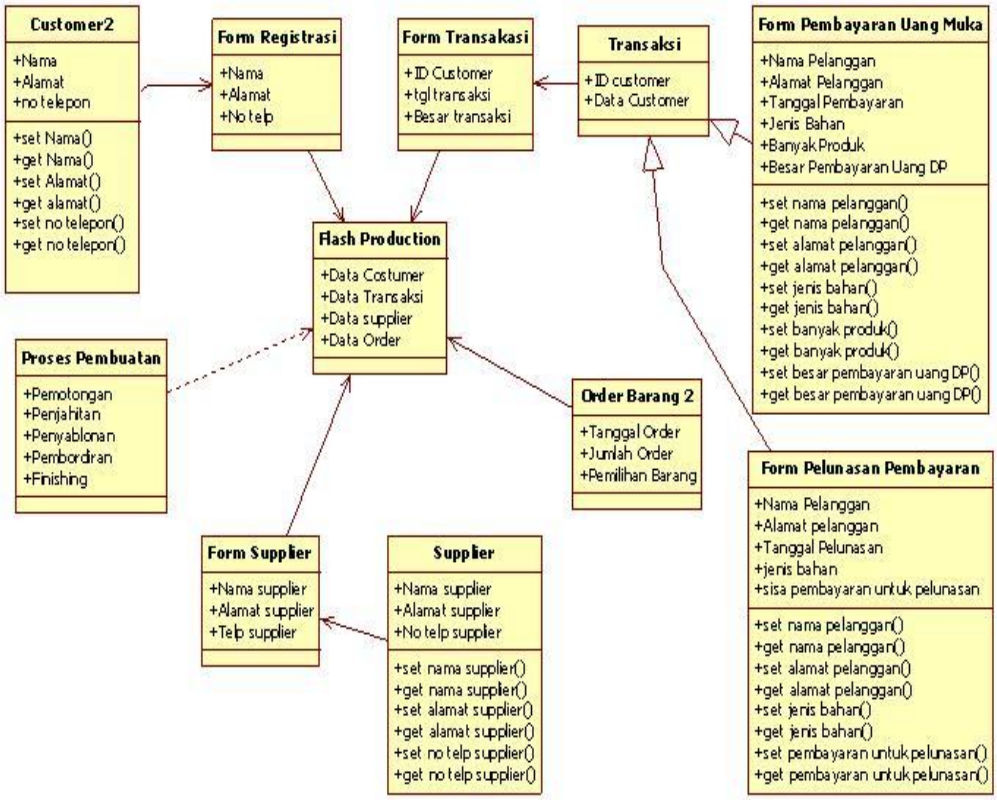
   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4) Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan coupling. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *choesion* yang kuat dan kadar *coupling* yang lemah.



Gambar 1.6 Class Diagram

Dalam class diagram terdapat beberapa relasi (hubungan antar class) yaitu;

1. Generalization dan Inheritence

   Diperlukan untuk memperlihatkan hubungan pewarisan (inheritance) antar unsur dalam diagram kelas. Pewarisan memungkinkan suatu kelas mewarisi semua atribut, operasi, relasi, dari kelas yang berada dalam hirarki pewarisannya.

2. Associations

   Hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui ekstensi class lain. Dalam notasi UML kita mengenal asosiasi 2 arah (bidirectional) dan 1 arah (undirectional).

3. Aggregation

   Hubungan antar-class dimana class yang satu (part class) adalah bagian dari class lainnya (whole class).

4. Composition

   Aggregation dengan ikatan yang lebih kuat. Di dalam composite aggregation, siklus hidup part class sangat bergantung pada whole class sehingga bila objek instance dari whole class dihapus maka objek instance dari part calss juga akan terhapus.

5. Depedency

   Hubungan antar-class dimana sebuah class memiliki ketergantungan pada class lainnya tetapi tidak sebaliknya.

6. Realization

   Hubungan antar-class dimana sebuah class memiliki keharusan untuk mengikuti aturan yang ditetapkan class lainnya. biasanya realization digunakan untk menspesifikasikan hubungan antara sebuah interface dengan class yang mengimplementasikan interface tersebut.

2.2.15 Design Pattern

Design Pattern (Pola Desain) adalah suatu bagian dari teknologi pemrograman berorientasi objek. Pola desain adalah solusi yang sudah teruji untuk menyelesaikan permasalahan pemrograman. Pola desain merupakan sebuah pola atau cara untuk mendesai komponen-komponen dalam pemrograman berorientasi objek yang baik sehingga komponen-komponen yang ada dapat digunakan kembali untuk aplikasi yang lain. Pola desain dimulai dengan menelusuri solusi-solusi yang baik dari suatu masalah untuk menghasilkan pola yang baik.

Cara membuat sebuah pola desain komponen yang baik adalah dengan memperhatikan kemungkinan perubahan sistem yang mungkin terjadi di masa mendatang. Pola desain penting untuk dipelajari agar dapat merancang program berorientasi objek yang dapat digunakan kembali untuk aplikasi lain dan memudahkan kerja tim jika sudah menggunakan pola pembuatan program

berorientasi objek yang sudah standar(mengurangi perbedaan cara berpikir antara anggota tim). Pola yang baik memperhatikan *cohesion* (keeratan keterkaitan metode dalam sebuah kelas) dan *coupling* (keeratan hubungan antar kelas).

Design Pattern (Pola Desain) sebaiknya dipelajari ketika pemrograman berorientasi objek telah dipahami. Pola desain bertujuan agar komponen program bersifat *reusable* (dapat digunakan kembali berulang-ulang). *Framework* merupakan rancangan arsitektur perangkat lunak yang digunakan sebagai standar pembuatan program dan memudahkan pengguna dalam memelihara perangkat lunak yang dibuat dengan menggunakan *framework*. *Framework* yang berorientasi objek adalah salah satu bagian implementasi pola desain. Submodul-submodul berikutnya akan membahas pola-pola desain yang telah ada.

Pada sebuah pola biasanya memiliki atribut-atribut (hal-hal yang perlu diperhatikan) sebagai berikut:

1) Nama (*Name*)
   Nama yang diberikan pada pola
2) Tujuan (*Intent*)
   Merupakan pemaparan singkat mengenai permasalahan yang terjadi dan tujuan yang ingin dicapai
3) Alias (*Also Known As*)
   Nama alias yang diberikan pada pola, nama alias bersifat opsional
4) Motivasi (*Motivation*)
   Sebuah skenario yang menjelaskan permasalahan perancangan dan solusi yang diberikan pola desain
5) Kemampuan diaplikasikan (*Applicability*)
   Spesifikasi situasi yang sesuai untuk penerapan pola
6) Struktur (*Structure*)
   Gambar sruktur yang menjelaskan hubungan kelas dan objek dalam pola
7) Partisipasi (*Participants*)
   Berbagai kelas atau objek yang berpartisipasi dan berperan dalam pola
8) Kolaborasi (*Collaborations*)

Bagaimana kolaborasi kerja sama dari partisipan untuk melaksanakan perannya masing-masing

9) Konsekuensi (*Consequences*)
   Bagaimana penapaian tujuan yang dilakukan pola dan konsekuensi penerapan pola
10) Implementasi (*Implementation*)
   Berbagai hal yang berkaitan dengan penerapan pola
11) Contoh Kode Program (*Sample Code*)
   Contoh kode program yang merepresentasikan penerapan pola
12) Contoh Penggunaan Pola (*Known Uses*)
   Contoh-contoh penggunaan pola pada sistem yang riil
13) Pola-pola Lain yang Terkait (*Related Pattern*)
   Pola-pola lain yang berhubungan dengan pola ini

Pola desain dibagi menjadi beberapa kelompok sebagai berikut :

1. Pola yang Berbasis pada Proses Pembuatan (*Creational Patterns*)
   Pola disusun dengan menitikberatkan komponen-komponen (objek-objek) apa saja yang perlu untuk dibuat
2. Pola yang Berbasis pada Struktur (*Structural Patterns*)
   Pola disusun dengan menitikberatkan pada struktur spesifikasi komponen-komponen (kesesuaian objek dengan tindakan atau kemampuan yang diharapkan) pemrograman berorientasi objek
3. Pola yang Berbasis pada Kebiasaan (*Behavioral Patterns*)
   Pola disusun dengan menitikberatkan pada spesifikasi kelakuan sistem aplikasi yang akan dibuat apakah pola yang dibuat sudah sesuai dengan kelakuan atau kemampuan aplikasi yang diharapkan.
4. Pola yang berbasis pada Desain Arsitektur (*Architectural Design Pattern*)
   Pola disusun dengan menitikberatkan pada hirarti arsitektur (pemisahan blok-blok lingkut kerja).

Model View Controller (MVC) merupakan salah satu contoh dari *Architectural Design Pattern.* Pada saat ini banyak *framework* yang didedikasikan pemrograman berorientasi objek.

Framework adalah kerangka kerja yang memudahkan *programmer* untuk membuat sebuah aplikasi sehingga *programmer* akan lebih mudah melakukan perubahan (customize) terhadap aplikasinya dan dapat memakainya kembali untuk aplikasi lain yang sejenis.

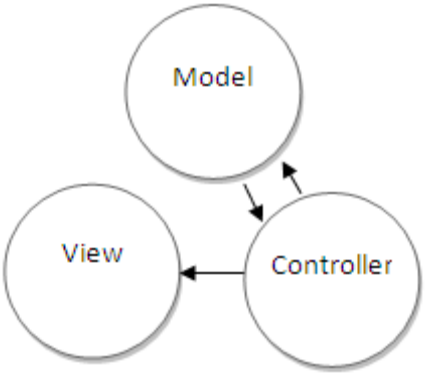
Contoh-contoh *framework* yang sering digunakan antara lain sebagai berikut

a) PHP : CodeIgniter, Smarty, cakePHP, Zend, Y2, dll
b) Java : Spring, Hibernate, Struts, dll
c) Ruby: Ruby on Rails, dll
d) Phyton: Django, dll
e) .NET: ASP.NET MVC Framework, dll

Konsep MVC bertujuan agar sebuah aplikasi dapat mudah dipelihara oleh orang-orang di dalam tim pengembangan yang berbeda spesifikasi pekerjaannya, misalnya *Database Administrator* (DBA) untuk mengurusi masalah basis data, blok *controller* untuk programmer, dan blok *view* untuk desainer antar muka (*interface designer*).

2.2.16 Model View Controller

Teknik pemrograman MVC (Model View Controller) merupakan teknik pemrograman yang populer saat ini, yang mengharapkan pemrogram secara disiplin untuk membagi program menjadi tiga bagian; model, view, dan controller, seperti gambar berikut :



Gambar 1.7 Arsitektur MVC

Model

Objek model adalah bagian dari aplikasi yang mengimplementasi logika untuk domain data aplikasi. Umumnya, objek model digunakan untuk mengambil data dari database atau menyimpan data ke dalam database.

View

View adalah komponen yang menampilkan antarmuka untuk pengguna (user interface, UI) aplikasi. Antarmuka ini dibuat berdasarkan data dari model.

Controller

Controller merupakan komponen yang digunakan untuk menangani interaksi pengguna, bekerja dengan model, dan memilih view mana yang digunakan untuk merender data.

Dalam aplikasi MVC, view hanya menampilkan informasi, controller menangani dan menanggapi masukan dan interaksi dengan pengguna. Dengan pola MVC, kita membuat aplikasi dengan memisahkan berbagai aspek dari aplikasi (logika input, bisnis, dan antarmuka), ada bagian penampil data (View), penyedia atau pemroses data (model), dan pengendali program (controller).

Dengan pemisahan ini, maka pembuatan satu program dapat dibagi-bagi kepada beberapa orang. Setiap bagian atau komponen yang dibuat secara terpisah diharapkan dapat digunakan tidak hanya oleh satu program saja, tetapi oleh banyak program, terutama bagian model dan view. Teknik pemrograman MVC ini d2mplementasi tidak hanya dalam PHP, tetapi suatu teknik pemrograman yang sudah dikenal dan digunakan hampir pada semua bahasa pemrograman yang ada.

2.3 Perangkat Lunak Pendukung

2.3.1 PHP

PHP adalah suatu bahasa *server-side* yang di desain khusus untuk aplikasi web. PHP dapat disisipkan diantara HTML, dan karena bahasa server-side, maka bahasa PHP akan dieksekusi di server, yang dikirimkan ke browser adalah “hasil jadi” bentuk HTML, dan kode PHP tidak akan terlihat.

PHP dahulu merupakan proyek pribadi dari Rasmus Lerdorf tahun 1994 (dengan dikeluarkannya PHP versi 1 yang digunakan untuk membuat home page pribadi yang dipublikasikan 8 Juni 1995). Versi pertama ini berupa kumpulan script PERL. Untuk versi kedua Ramus menilis ulang script-script PERL tersebut menggunakan bahasa C, kemudian menambahkan fasilitas untuk form HTML dan koneksi MYSQL. Adapun PHP didapat dari singkatan *PHP Hypertex Prepocessor*.

Setelah mengalami perkembangan, Zeev Suraski dan Andi Gumans (termasuk Rasmus) yang bergabung dalam suatu kelompok open source, maka mereka memulai versi 3 PHP nya, PHP telah menampakkan keunggulan sebagai salah satu bahasa *server scripting* yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini.pada bulan Mei 2000 dirilis PHP versi 4 yang dikeluarkan oleh Zend Engine yaitu sebuah produk yang diciptakan oleh Suraski dan Gutmans. Kehandalan PHP diperlengkap dengan dirilisnya PHP versi 5 pada 13 Juli 2004 yang mendukung OOP serta berbagai manajemen database yang luas dan penanganan error yang mudak. Sintak yang digunakan berasal dari bahasa C, Java maupun PERL. Sampai tulisan ini dibuat versi PHP yang terbaru adalah versi 5.4.7 yang dirilis pada tanggal 13 September 2012.

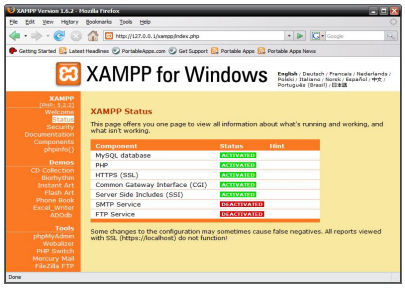
```
<?php if ($xyz>=3)
{Print $myHeading;}
Else {?>
DEFAULT HEADING
<?php } ?>
```

PHP merupakan bahasa script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu terbaru. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. Oleh karena itu, spesifikasi server lebih berpengaruh pada eksekusi dari script php daripada spesifikasi client. Namun tetap

diperhatikan bahwa web yang dihasilkan tentunya harus dapat dibuka oleh browser pada client. Dalam hal ini versi dari html yang digunakan harus didukung oleh browser client. PHP termasuk dalam *Open Source Product.* Jadi dapat menambah *source code* dan dapat mendistribusikannya secara bebas.

2.3.2 XAMPP

XAMPP adalah salah satu paket instalasi Apache, PHP, dan MySQL secara instan yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*) [6].



Gambar 1.8 XAMPP

Fitur yang ada pada XAMPP adalah sebagai berikut :

1. Apache
2. Cgi-Bin
3. PHP
4. MySQL
5. FTP
6. *Mercury Mail* (SMTP)
7. PHP MyAdmin
8. Perl

2.3.3 Macromedia Dreamweaver

Pengertian *Macromedia Dreamweaver* adalah sebuah *software* HTML *editor* profesional yang digunakan untuk mendesain secara visual dan mengelola situs web maupun halaman web. Bilamana kita menyukai untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan *editing*, *Dreamweaver* membuatnya menjadi lebih

mudah dengan menyediakan tools yang sangat berguna dalam peningkatan kemampuan dan pengalaman kita dalam mendesain web.



Gambar 1.9 Macromedia Dreamweaver

Teknologi *Dreamweaver Roundtrip* HTML mampu mengimpor dokumen HTML tanpa perlu memformat ulang kode tersebut dan kita dapat menggunakan *Dreamweaver* pula untuk membersihkan dan memformat ulang HTML bila kita menginginkannya. Selain itu *Dreamweaver* juga dilengkapi kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam situs. Kita juga dapat melakukan evaluasi situs dengan melakukan pengecekan *broken link*, kompatibilitas *browser*, maupun perkiraan waktu *download* halaman web.

2.3.4 MYSQL

Database MySQL merupakan sistem manajemen basisdata SQL yang sangat terkenal dan bersifat open source. MySQL dibangun, didistribusikan, dan didukung oleh MySQL AB. MySQL AB merupakan perusahaan komersial yang dibiayai oleh pengembang MySQL. MySQL mempunyai dua maca lisensi yaitu lisensi yang bersifat open source dengan menggunakan GNU (*General Public License*) dan lisensi kedua berupa *Standard Commercial License* yang dapat dibeli dari MySQL AB.

Dalam konteks bahasa SQL, informasi disimpan dalam tabel-tabel yang secara logis merupakan struktur dua dimensi yang tersimpan atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column)*. Baris pada tabel disebut *insance* dari data sedangkan kolom sering disebut sebagai *atributes* atau *field.*

MySQL mempunyai logo berupa ikan lumba-lumba yang bernama Sakila. Nama tersebut merupakan kontribusi dari pemenang kontes “Nama Lumba-lumba” yang diadakan oleh MySQL AB. Pemenang kontes tersebut bernama Ambrozze Twebaze, seorang developer software open source dari Swaziland, Afrika. Menurut Ambroze nama Sakila berasal dari SISwati bahasa lokal Swaziland. Sakila juga merupakan nama kota di Arusha Tanzania.

MySQL yang berjalan di sistem operasi Windows terbukti sangat stabil. MySQL versi Windows mempunyai fitur yang sama dengan MySQL versi Unix.

Sintak SQL pada MySQL :

1. *CREATE TABLE*

   Digunakan untuk membuat tabel baru.

```
CREATE TABLE nama_tabel (nama_kolom tipekolom(size), nama_kolom tipe kolom(size), … ) ;
```

2. *SELECT*

   Berfungsi untuk menampilkan data dari tabel.

```
SELECT nama_kolom, nama_kolom, … FROM nama_tabel WHERE kriteria;
```

3. *UPDATE*

   Digunakan untuk memperbaiki informasi atau data pada table.

```
UPDATE nama_tabel SET nama_kolom = value WHERE kriteria;
```

4. *DELETE*

   Untuk menghapus record atau data dari table.

```
DELETE FROM nama_tabel WHERE kriteria;
```

5. *INSERT INTO*

   Untuk menambah informasi/data baru ke dalam tabel.

```
INSERT INTO nama tabel (kolom) VALUES (‘data1’, data2
```

```
    …);
```

6. *CREATE DATABASE*

   Digunakan untuk membuat database baru.

```
CREATE DATABASE nama_database;
```

   (nama database bebas berdasarkan kasus)

7. *DROP DATABASE*

   Digunakan untuk menghapus database.

```
DROP DATABASE nama_database;
```

   (nama database bebas berdasarkan kasus)

8. *DROP TABLE*

   Digunakan untuk menghapus tabel.

```
DROP TABLE nama_table;
```

9. *RENAME*

   Untuk mengganti nama pada tabel.

```
RENAME nama_tabel_lama TO nama_tabel_baru;
```

10. *DESC TABLE*

    Deskripsi tabel atau kolom.

```
DESCRIBE | desc nama_table [nama_kolom]
```

11. *ALTER TABLE and ADD*

    Untuk menambahkan kolom atau field.

```
ALTER TABLE nama_tabel ADD nama_kolom tipe
   kolom(size);
```

12. *ALTER TABLE and CHANGE*

    Untuk mengganti nama kolom atau field.

```
ALTER TABLE nama_tabel CHANGE nama_kolom_yang_diganti
nama_kolom_baru tipe kolom baru(size);
```

13. *ALTER TABLE and MODIFY*

Untuk mengganti type atau size pada kolom atau field.

```
ALTER TABLE namatabel MODIFY nama_kolom tipe kolom(size);
```

14. *GRANT*

Memberikan akses kepada user terhadap table, bisa juga digunakan untuk membuat user baru.

```
GRANT (all pr4ileges | (select, insert, update,delete, references(column list), usage)) on table to user, … [with grant option]
```

15. *LOCK TABEL*

Untuk menutup atau mengunci akses user pada table.

```
LOCK TABLE nama_table [as alias] read | write [table_name read | write]
```

16. *UNLOCK TABLE*

Untuk membuka akses pada table yang sebelumnya dikunci.

```
UNLOCK nama_table
```

17. *LOAD DATA INIFILE*

Perintah untuk membaca data dari file teks.

```
load data inifile syntax
```

2.3.5 CodeIgniter

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis. Pemrogram tidak perlu membuat program dari awal (*from scracth*), karena CI

menyediakan sekumpulan librari yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengkakses librarynya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan.

Framefork CI merupakan framework yang memiliki dokumentasi yang jelas dan lengkap, yang memudahkan pengembang untuk mempelajari dengan mudah. Pendekatan dari CI sangatlah mudah, dari membuat sekadar tulisan sampai dengan yang kompleks dapat didekati dengan mudah. Tidak seperti framework yang lain, untuk mendapatkan tulisan Hello World di browser saja, kita harus menggunakan beberapa tahap. CI cukup dengan satu file dan satu prosedur atau method.

Framework CI telah dibahas juga dalam forum di IBM dan Oracle untuk pengembangan aplikasi berbasis web dengan menggunakan PHP. Seperti telah diketahui, IBM dan Oracle telah memiliki kerjasama dengan Zend, yang mengembangkan PHP, dengan produk yang dikenal dengan ZendCore for IBM DB2 dan Informix, dan ZendCore for Oracle. Produk ZendCore dipesan khusus oleh kedua perusahaan tersebut, untuk memudahkan pengembangan aplikasi dengan PHP menggunakan database IBM DB2 atau Informix, dan Oracle. Pengembang tidak harus direpotkan atau sulit memasang PHP agar bisa mengakses database-database tersebut.

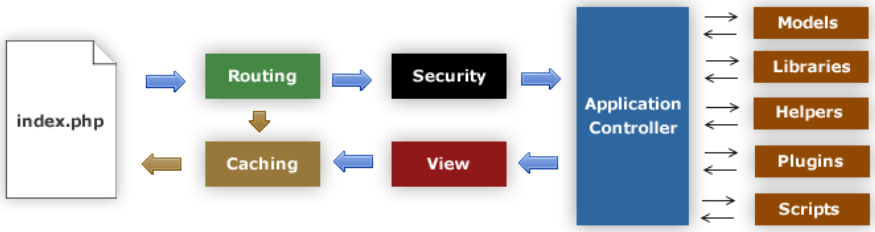
CI dapat digunakan dengan menggunakan server web Apache dan MS ISS> hanya saja, tidak semua fitur dari CI dapat diimplementasikan sempurna di MS ISS, karena CI dirancang dan dikembangankan dengan acuan Apache; contoh yang pasti adalah penggunaan file .htaccess, yang memang milik Apache. Secara umum, maka CI memberikan dan menyediakan :

1) Suatu kerangka kerja dengan jejak yang sedikit (*small footprint*).
2) Kinerja yang bagus.
3) Kompatibilitas dengan akun hosting standar yang terpasang dengan perbedaan versi dan konfigurasi.
4) Suatu kerangka kerja yang memerlukan hampir tidak memerlukan konfigurasi.

5) Suatu kerangka kerja yang tidak memerlukan kita menggunakan baris perintah (*command line*).
6) Suatu kerangka kerja yang tidak mengharuskan bekerja dengan aturan pengkodean yang mengikat/membatasi.
7) Suatu aturan yang tidak mengharuskan kita dipaksa menggunakan suatu bahasa *templating* (walaupun CI menyediakan suatu parser untuk *template*).
8) Cara untuk menghindarkan kompleksitas dalam membuat program, dengan menyediakan solusi yang sederhana.
9) Suatu dokumentasi kerangka kerja yang menyeluruh dan jelas.

2.3.5.1 Pemrograman MVC dengan CodeIgniter

CI adalah satu framework pemrograman PHP yang mengimplementasikan teknik pemrograman MVC. Berikut adalah arsitektur bagaimana CI mengimplementasikan MVC.



Gambar 1.10 Arsitektur CodeIgniter

Index.php berlaku sebagai controller di depan, yang menginisialisasi sumber daya yang dibutuhkan untuk menjalankan CodeIgniter

1) Router akan memeriksa request HTTP untuk menentukan apa yang harus dilakukan dengan request tersebut.
2) Jika ada file cache, maka file cache langsung dikirimkan kepada browser, dengan memotong eksekusi sistem normal.
3) Keamanan. Sebelum suatu aplikasi controller dipanggil, request HTTP dan data yang dikirimkan oleh pengguna disaring (filter) terlebih dahulu untuk keamanan.
4) Controller akan memuat model, libari inti, helper, dan sumberdaya lainnya yang diperlukan untuk memproses request tersebut.

5) View terakhir direnderkemudian dikirimkan kepada browser untuk dilihat. Jika caching didefinisikan, maka view akan dicache terlebih dahulu, setelah itu baru dikirimkan kepada browser.

Codeigniter (CI) bisa digunakan untuk belajar melakukan pemrograman dengan teknik pemrograman yang menggunakan konsep MVC (Model View Controller). CI mengijinkan pemrogram untuk bekerja penuh di dalam controller, sambil memahami konsep MVC dengan benar.

Secara umum, kita harus membuat tiga file yang menyatakan tiga komponen; file mode, view dan file controller untuk bisa membuat sebuah aplikasi. Dalam pemrograman MVC dengan menggunakan CI, kita tidak harus membuat tiga file, di awal pembuatan aplikasi dengan CI, untuk bisa menyesuaikan diri sebelum akhirnya menggunakan sedikitnya tiga file untuk bisa menyelesaikan satu modul.

2.4 Teori Kuesioner

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner merupakan teknik pengumpulan data yang efisien bila peneliti tahu dengan pasti variable yang akan diukur dan tahu apa yang bisa diharapkan dari responden. Selain itu, kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar diwilayah yang luas. Kuesioner dapat berupa pertanyaan atau pernyataan tertutup atau terbuka, dapat diberikan kepada responden secara langsung atau dikirim melalui POS atau internet.

Bila penelitian dilakukan pada lingkup yang tidak terlalu luas, sehingga kuesioner dapat diantarkan langsung dalam waktu yang tidak terlalu lama, maka tidak perlu dikirm melalui POS. Dengan adanya kontak langsung antara peneliti dengan responden akan menciptakan suatu kondisi yang cukup baik, sehingga responden dengan sukarela akan memberikan data obyektif dan cepat.

Dalam buku Prof. Dr. Sugiyono [7] mengemukakan beberapa prinsip dalam penulisan angket sebagai teknik pengumpulan data yaitu : prinsip penulisan, pengukuran, dan penampilan fisik.

1. Prinsip Penulisan Angket

   Prinsip ini menyangkut beberapa factor yaitu : isi dan tujuan pertanyaan, bahasa yang digunakan mudah, pertanyaan terbuka terutup-negatif positif, pertanyaan tidak mendua, tidak menanyakan hal-hal yang sudah lupa, pertanyaan tidak mengarahkan, panjang pertanyaan dan urutan pertanyaan.

2. Prinsip Pengukuran

   Angket yang diberikan kepada responden adalah merupakan instrumen penelitian, yang digunakan untuk mengukur variabel yang akan diteliti. Oleh karena itu instrument angket tersebut harus dapat digunakan untuk mendapatkan data yang valid dan reliabel tentang variabel yang diukur.

3. Penampilan Fisik Angket

   Penampilan fisik angket sebagai alat pengumpulan data akan mempengaruhi respon atau keseriusan responden dalam mengisi angket. Angket yang dibuat di kertas buram, akan mendapat respon kurang menarik bagi responden, bila dibandingkan dengan kertas yang bagus dan berwarna. Tetapi angket yeng dicetak di kertas yang bagus berwarna akan menjadi mahal dalam pembiayaan.

2.4.2 Skala Likert

Skala *Likert* [7] digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti yang selanjutnya disebut sebagai variabel penelitian. Dengan skala *Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel.

Kemudian indicator tersebut dijadikan sebagai titik tolak untuk menyusun item item instrumen yang dapat berupa pernyataan atau pertanyaan.Jawaban setiap item instrumen yang menggunakan skala *Likert* mempunyai gradasi dari sangat positif sampai sangat negatif, yang berupa kata-kata antara lain :

Tabel 1.3 Jawaban Skala Likert

No	Jawaban	No	Jawaban
1	sangat setuju	1	setuju

2	setuju	2	sering
3	ragu-ragu	3	kadang-kadang
4	tidak setuju	4	hampir tidak pernah
5	sangat tidak setuju	5	tidak pernah
No	**Jawaban**	**No**	**Jawaban**
1	sangat positif	1	baik sekali
2	positif	2	cukup baik
3	netral	3	kurang baik
4	negatif	4	sangat tidak baik
5	sangat negatif		

Untuk keperluan analis kuantitatif, maka jawaban itu dapat diberi skor, misalnya :

Tabel 1.4 Jawaban Analis Kuantitatif

No	Jawaban	Score
1	Setuju/selalu/sangat positif	5
2	Setuju/sering/positif	4
3	Ragu-ragu/kadang-kadang/netral	3
4	Tidak setuju/hampir tidak pernah/negative	2
5	Sangat tidak setuju/tidak pernah/sangat positif	1

Instrumen dalam penelitian yang menggunakan metoda skala *Likert* dapat dibuat pertanyaan atau pernyataan-nya dalam bentuk checklist ataupun pilihan ganda.

a. Contoh bentuk checklist

   Berilah jawaban pertanyaan berikut sesuai engan pendapat anda, dengan cara memberi tanda (✔) pada kolom yang tersedia.

Tabel 1.5 Contoh Bentuk Ceklis

No.	Pertanyaan	2. Jawaban				
		SS	ST	RG	TS	STS
1.	Apakah anda setuju, sistem informasi yang dibuat ini mampu memudahkan dan mengatasi masalah yang ada sebelumnya		✔			
2.	……					

Kemudian dengan teknik pengumpulan data angket, maka instrumen tersebut diberikan kepada 100 orang pegawai yang diambil secara acak. Dari 100 orang pegawai setelah dilakukan analisis misalnya

Tabel 1.6 Keterangan

Keterangan :	
SS	= Sangat setuju
ST	= Setuju
RG	= Ragu-ragu
TS	= Tidak seutuju
STS	= Sangat tidak setuju
Bila :	
SS	diberi skor = 5
ST	diberi skor = 4
RG	diberi skor = 3
TS	diberi skor = 2
STS	diberi skor =

Berdasarkan data tersebut 65 orang atau 65% pegawai menjawab setuju dan sangat setuju. Jadi kesimpulannya mayoritas pegawai setuju dengan adanya sistem informasi.

Data tersebut juga dapat dianalisis berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan maka :

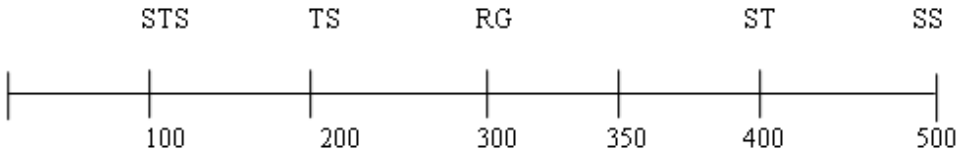
Tabel 1.7 Skoring Jawaban

25 orang pegawai menjawab SS
40 orang pegawai menjawab ST
5 orang pegawai menjawab RG
20 orang pegawai menjawab TS
10 orang pegawai menjawab STS

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 100 = 500 (SS) skor tertinggi jumlah skor rendah 1 x 100 = 100 (STS).

Jadi berdasarkan data itu maka tingkat persetujuan terhadap sistem informasi yang dibuat = (350 : 500) x 100% = 70%

Secara kontinum dapat digambarkan seperti berikut :



Gambar 1.11 Jumlah Scoring

Jadi berdasarkan data yang diperoleh dari 100 responden maka data 350 terletak pada daerah setuju.

Bila berdasarkan pada kelompok responden, maka dapat diketahui bahwa :

Tabel 1.8 Tabel Jumlah Responden

1	52 prosen menyatakan sangat setuju (25/100 = 25%)
2	40 prosen menyatakan setuju (40/100 = 40%)
3	5 prosen menyatakan ragu-ragu (5/100 = 5%)
4	20 prosen menyatakan tidak setuju (20/100 = 20%)
5	10 prosen menyatakan sangat tidak setuju (10/100 = 10%)

b. Contoh bentuk pilihan ganda.

Berilah satu jawaban terhadap pertanyaan berikut sesuai dengan pendapat anda, dengan member tanda lingkaran pada nomor jawaban yang tersedia.

1. Apakah anda setuju, sistem informasi yang dibuat mampu memudahkan dan mangatasi masalah yang ada sebelumnya ?
   a. Sangat tidak setuju
   b. Tidak setuju
   c. Ragu-ragu
   d. Setuju
   e. Sangat setuju

Akan tetapi dengan bentuk pilihan ganda itu, maka jawaban dapat diletakkan pada tempat yang berbeda-beda. Semisal untuk jawaban diatas “sangat setuju” diletakkan di paling akhir untuk selanjutnya jawaban “sangat tidak setuju diletakkan di paling awal. Dalam penyusunan instrumen untuk variabel tertentu, sebaiknya butir butir pertanyaan dibuat dalam bentuk kalimat positif, netral, ataupun negative sehingga responden dapat menjawab dengan serius dan konsisten.

Dengan cara demikian maka kecendrungan responden untuk menjawab kolom tertentu dari bentuk checklist dapat dikurangi. Dengan model ini juga selalu membaca pertanyaan setiap item instrumen dan juga jawabannya. Pada bentuk checklist, sering jawaban tidak dibaca, karena letak jawaban sudah menentu. Tetapi dengan bentuk checklist juga akan didapat keuntungan dalam hal ini singkat dalam pembuatannya, hemat kertas, mudah mentabulasikan data, dan secara visual lebih menarik. Data yang diperoleh dari skala tersebut adalah berupa interval.

2.4.3 Teori Pengujian Sistem

Pengujian sistem dimaksudkan untuk menunjukkan bahwa program melakukan apa yang dimaksudkan dan untuk menemukan cacat program sebelum mulai digunakan. Ketika pengguna menguji perangkat lunak, pengguna menjalankan sebuah program dengan menggunakan data buatan. pengguna memeriksa hasil uji coba untuk mengetahui kesalahan, anomali, atau informasi

tentang atribut non-fungsional program. Proses pengujian memiliki dua tujuan yang berbeda :

1. Untuk menunjukkan kepada pengembang dan pelanggan bahwa perangkat lunak memenuhi persyaratan. Untuk perangkat lunak kustom, perangkat lunak ini harus ada setidaknya satu tes untuk setiap kebutuhan dalam dokumen persyaratan. Untuk produk perangkat lunak generik, harus ada tes untuk semua fitur sistem, ditambah kombinasi dari fitur ini, yang akan dimasukkan dalam rilis produk.
2. Untuk menemukan situasi di mana perilaku perangkat lunak tidak benar, tidak diinginkan, atau tidak sesuai dengan spesifikasinya. Ini adalah konsekuensi dari cacat perangkat lunak. Pengujian cacat berkaitan dengan membasmi perilaku sistem yang tidak diinginkan seperti sistem *crash*, interaksi yang tidak diinginkan dengan sistem lain, perhitungan yang salah, dan korupsi data.

Tujuan pertama mengarah ke pengujian validasi, di mana pengguna mengharapkan sistem untuk berjalan dengan benar menggunakan satu set kasus uji yang mencerminkan harapan pengguna terhadap sistem. Tujuan kedua mengarah ke pengujian cacat, di mana uji kasus dirancang untuk mengekspos cacat. Uji kasus dalam pengujian cacat dapat sengaja disalahkan dan tidak perlu mencerminkan bagaimana sistem biasanya digunakan. Tentu saja, tidak ada batasan yang jelas antara kedua pendekatan ini untuk pengujian. Selama pengujian validasi, pengguna akan menemukan cacat dalam sistem, selama pengujian cacat beberapa tes akan menunjukkan bahwa program tersebut memenuhi persyaratan. Pengujian tidak dapat menunjukkan bahwa perangkat lunak bebas dari cacat atau akan berperilaku seperti yang telah ditentukan [6].