BAB 2
TINJAUAN PUSTAKA

2.1 Profil Studi Kasus

Merupakan uraian mengenai tempat penelitian yang dilakukan dari studi kasus. Profil yang akan diuraikan terdiri dari sejarah, logo, visi, misi perusahaan dan struktur organisasi. Referensi yang digunakan sebagai acuan untuk menjelaskan profil PT. Kalacakra Teknologi berasal dari *Company Profile* yang dibuat oleh perusahaan yang bersangkutan.

PT. Kalacakra Teknologi merupakan perusahaan yang bergerak di bidang penyedia jasa aplikasi (*Software Vendor*) dan konsultan IT yang memfokuskan diri kepada solusi perdagangan saham beserta turunannya yang beralamat di Gedung DBS Bank Tower Lantai 28 Ciputra World Jakarta, Jl. Prof. Dr. Satrio Kaveling 3-6 Jakarta, 12940.

2.1.1 Sejarah PT. Kalacakra Teknologi

PT. Kalacakra Teknologi merupakan *software vendor* yang berfokus pada membangun perangkat aplikasi untuk industri pasar modal. Aplikasi tersebut digunakan oleh perusahaan-perusahaan sekuritas (*broker*) yang terdaftar secara resmi di Bursa Efek Indonesia (BEI). PT Kalacakra Teknologi didirikan pada Juli 2009, merupakan perusahaan penyedia jasa aplikasi (*Software Vendor*) dan konsultan IT yang memfokuskan diri kepada solusi perdagangan saham beserta turunannya. Dengan percaya diri PT. Kalacakra Teknologi dapat menyediakan solusi yang "*Comprehensive*", "*Robust*", dan "*Proven*" serta dapat diimplementasikan pada setiap Perusahaan secara "*Scalable*" dan "*Flexible*". Berbagai keahlian yang dimiliki pegawai PT. Kalacakra Teknologi yaitu *business consultant, solution architect, project management, application development, databases, Capital Market Industry expertise, etc*.

Pada area ini, PT. Kalacakra Teknologi mengeksploitasi kemampuan yang lebih luas terhadap *development platform*. Untuk kebutuhan *Stock Market*, PT. Kalacakra Teknologi menyediakan solusi JONEC*, Front-Office, Back-Office*, N-

Transform (*Regulator Report*), dan *Engine Gateway* untuk kebutuhan *trading* yang menghubungkan *stock brokers* dengan IDX (Indonesia *Stock Exchange*) melalui *channel* yang *secure*. BizEquity adalah solusi *stock brokerage system*. BizEquity memungkinkan *stock broker* untuk bertransaksi secara langsung ke IDX (Indonesia *Stock Exchange*) *scriptless trading system.* PT. Kalacakra Teknologi juga menjalin kerjasama strategis dengan IMQ – LKBN Antara untuk menyediakan *stock*/*market* info melalui sistem PT. Kalacakra Teknologi. PT. Kalacakra Teknologi memberikan layanan kepada berbagai perusahaan seperti, PERUSAHAAN SEKURITAS, CIMB *Securities*, PT. Bumiputera *Capital* Indonesia, Indo Premier *Securities*, Bursa Efek Indonesia, Phillip *Securities*, Deutsche Bank, DBS Vickers *Securities*, JPMorgan, Morgan Stanley, dan UBS.

Beberapa produk yang sudah dikembangkan dengan baik dan telah diimplementasikan dibanyak tempat, seperti :

1. BizEquity/oms : *Equity Order Management System*
2. N-FIX Engine : *Multi Channel FIX Engine Application*
3. N-MIFeed : *Market Info Processing System*
4. Nayaga Middle Office : *Equity Middle Office System* OMGEO *Integration*
5. Nayaga Back Office : *Brokerage Back Office System*
6. N-Transform : ELT *Engine for Capital Market Regulatory Report*

2.1.2 Visi dan Misi PT. Kalacakra Teknologi

Visi adalah sebuah gambaran mengenai tujuan dan cita-cita di masa depan yang harus dimiliki organisasi sebelum disusun rencana bagaimana mencapainya. Misi adalah pernyataan luas dan kekal mengenai nilai suatu perusahaan. Tanpa adanya visi dan misi sebuah perusahaan tidak akan bertahan dan tidak tahu apa yang harus dilakukan oleh perusahaan tersebut [1].

PT. Kalacakra Teknologi memiliki visi dan misi sebagai berikut :

1. Visi PT. Kalacakra Teknologi

Untuk Menjadi perusahan pemberi solusi IT terbaik pada *client*.

2. Misi PT. Kalacakra Teknologi

a. Memfokuskan diri kepada solusi perdagangan saham beserta turunannya.

b. Menciptakan produk-produk solusi IT terbaik untuk menjawab permasalahan pelanggan.
c. Dapat menyediakan solusi yang "*Comprehensive*", "*Robust*", dan "*Proven*" serta dapat diimplementasikan pada setiap Perusahaan secara "*Scalable*" dan "*Flexible*".

2.1.3 Logo PT. Kalacakra Teknologi

Logo merupakan sebuah simbol yang dirancang untuk mewakili karakter dan menjadi identitas dari sebuah perusahaan. Logo merupakan bentuk ekspresi dan bentuk visual dari konsepsi perusahaan, produk, organisasi, maupun institusi serta merupakan simbol visual yang memiliki bentuk yang berasal dari nilai strategis perusahaan yang bersangkutan. Sebagai bagian identitas perusahaan, logo ibarat bagian tubuh yang mampu mengutarakan isi hati produk atau perusahaan. Dari sisi pemasaran, logo mempunyai fungsi pembeda produk dengan produk lainnya. Pada dasarnya logo, jika dikategorikan berdasarkan unsur pembentuknya terdiri dari *logotype*, *logogram*, dan gabungan antara *logotype* dan *logogram* [2].



Gambar 2.1 Logo PT. Kalacakra Teknologi

Arti Logo PT. Kalacakra Teknologi :

1) *Logotype* (Cakra *Technology*) tersebut merupakan kependekan dari PT. Kalacakra Teknologi.
2) Bentuk K diganti dengan panah pada *logotype* melambangkan kecepatan, keakuratan dan pentingnya perkembangan ke depan dan menandakan bahwa

PT. Kalacakra Teknologi meningkat menjadi perusahaan terpercaya bersama semua anggota dan *investor*.

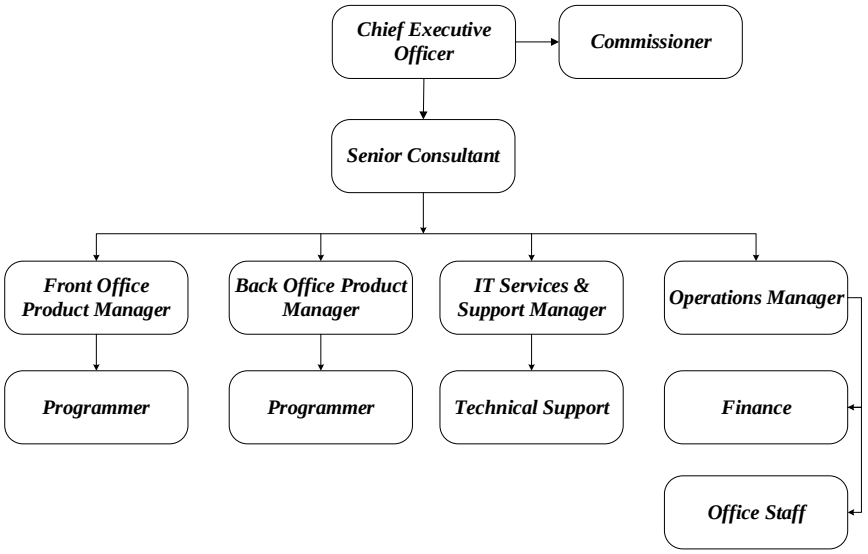
3) Warna hijau pada *logotype* melambangkan pertumbuhan, muda, kesuksesan materi, pembaharuan, daya tahan, keberuntungan, dan kesehatan, kehidupan, sifat positifnya, optimisme, kebebasan, keseimbangan, ketabahan.
4) Warna abu-abu tua pada *logotype* melambangkan keseimbangan, keamanan, kepandaian, tenang dan serius, profesional, kualitas.
5) Warna abu pada *logotype* melambangkan keseriusan, bisa diandalkan dan stabil.

2.1.4 Badan Hukum PT. Kalacakra Teknologi

PT. Kalacakra Teknologi sudah berbadan hukum PT (Perseroan Terbatas), yang disahkan pada tahun 2015. Berdasarkan SK No. AHU-0004480.AH.01.02. Tahun 2015, Notaris Irnova Yahya, SH.

2.1.5 Struktur Organisasi PT. Kalacakra Teknologi

Pada dasarnya suatu perusahaan perlu membuat struktur organisasi, dimana menjelaskan secara rinci mengenai tugas, tanggung jawab, dan wewenang masing-masing jabatan yang terdapat di struktur organisasi tersebut. Struktur organisasi sangat penting agar kerjasama antara masing-masing jabatan dapat terjalin secara harmonis dan dapat mewujudkan perusahaan dalam pencapaian visi dan misinya.



Gambar 2.2 Struktur Organisasi PT. Kalacakra Teknologi

2.1.6 Job Description PT. Kalacakra Teknologi

Berikut adalah deskripsi pekerjaan dari struktur organisasi PT. Kalacakra Teknologi :

1. *Chief Executive Officer*
   a. Mengatur, mengelola dan memimpin perusahaan secara umum.
   b. Seorang pimpinan yang bertanggung jawab atas kegagalan atau kesuksesan sebuah perusahaan.
   c. Menentukan kebijakan terkait perusahaan.
   d. *Visioner*, memastikan perusahaan mencapai misi yang ditentukan.
2. *Commissioner*
   a. Memastikan keberadaan dan kecukupan sumber keuangan.
   b. Memilih eksekutif puncak perusahaan, yang dikepalai oleh CEO.
   c. Mengesahkan anggaran tahunan.
   d. Menetapkan dan memperbaiki misi perusahaan.
   e. Menentukan gaji dan kompensasi mereka sendiri.
3. *Senior Consultant*
   a. Memberikan konsultansi/rekomendasi mengenai solusi IT terbaik untuk memecahkan masalah.
   b. Memberikan konsultasi kepada *client*/*customer* dan lebih banyak berhadapan dengan banyak orang.
   c. Membuat dokumen seperti proposal, *requirement* dan desain *software* secara umum.
4. *Front & Back Office Product Manager*
   a. Sebagai "pengasuh" dari produknya agar produknya itu tetap dapat mengatasi *problem* dari para pelanggannya.
   b. Memastikan bahwa segala promosi dan aktivitas yang dilakukan tetap menguntungkan bagi perusahaannya.
   c. Menjadi *translator* visi perusahaan ke dalam *product* yang dibuat.
   d. Membuat *relevant product* yang sesuai dengan masalah yang dihadapi oleh *user*.

5. *IT Services & Support Manager*
    a. Menerima, memprioritaskan dan menyelesaikan permintaan bantuan IT.
    b. Membeli *hardware* IT, *software* dan hal-hal lain yang berhubungan dengan hal tersebut.
    c. Instalasi, perawatan dan penyediaan dukungan harian baik untuk *hardware* & *software* Windows & Macintosh, peralatan termasuk *printer*, *scanner*, *hard-drives external*, dan lain-lain.
    d. Korespondensi dengan penyedia jasa eksternal termasuk *Internet Service Provider*, penyedia jasa Email, *hardware*, dan *software supplier*, dan lain-lain.
6. *Operations Manager*
    a. Memantau dan menjaga pengeluaran biaya sesuai dengan anggaran yang telah ditetapkan oleh perusahaan .
    b. Mengelola, mengawasi, dan mengkoordinasi seluruh kegiatan opersaional perusahaan.
7. *Programmer*
    a. Membangun/mengembangkan *software* terutama pada tahap *construction* dengan melakukan *coding* dengan bahasa pemprograman yang ditentukan.
    b. Mengimplementasikan *requirement* dan desain proses bisnis ke komputer dengan menggunakan algoritma /logika dan bahasa pemprograman.
    c. Melakukan *testing* terhadap *software* bila diperlukan.
8. *Technical Support*
    a. Menangani *downtime server.*
    b. Pemantauan performa *server.*
    c. *Troubleshooting* fitur cPanel problematik.
    d. Melakukan *remote hosting client* untuk penanganan lebih dalam.
9. *Finance*
    a. Bertanggung jawab atas pelaporan keuangan secara keseluruhan dan prosedur fiskal, sumber daya manusia dan kegiatan administrasi sehari-hari.
    b. Melakukan transaksi keuangan perusahaan.
    c. Melakukan penagihan kepada *customer.*

d. Mencatat, memonitor dan mengontrol seluruh pemasukan dan pengeluaran perusahaan.

   Menyiapkan dokumen penagihan *invoice*/kwitansi tagihan beserta kelengkapannya.

10. *Office Staff*

    a. Menyiapkan urusan logistik, seperti mengawasi *transport*, ketersediaan tempat penyimpanan barang, penyediaan ATK (alat tulis kantor), peralatan kantor, dan lain-lain.

2.1.7 Proses Bisnis Perusahaan Sekuritas

1) Persyaratan yang harus dipenuhi oleh calon Nasabah

   1. Mengisi dan menandatangani aplikasi Pembukaan Rekening Efek dari Perusahaan Sekuritas bersangkutan.
   2. Melampirkan Surat Kuasa Pembukaan Rekening Efek di PT. Kustodian Sentral Efek Indonesia (KSEI) yang disediakan oleh Perusahaan Sekuritas bersangkutan.
   3. Menyerahkan dokumen yang dipersyaratkan oleh Perusahaan Sekuritas bersangkutan, yaitu:
      a) Fotokopi kartu indentitas (KTP atau Passport)
      b) Fotokopi Nomor Pokok Wajib Pajak (NPWP)

2) Syarat-Syarat Dan Ketentuan-Ketentuan Mengenai Rekening Efek

   1. Penunjukan Perusahaan Sekuritas

   1.1. Nasabah dengan ini mengajukan permohonan untuk membuka Rekening Efek pada Perusahaan Sekuritas serta menunjuk Perusahaan sekuritas untuk bertindak selaku perantara untuk setiap transaksi Efek yang dilakukan oleh Nasabah di Bursa Efek dengan syarat-syarat dan kondisi yang ditetapkan oleh Perusahaan sekuritas.

   1.2. Nasabah setuju bahwa Perusahaan sekuritas mempunyai hak sepenuhnya untuk menerima atau menolak permohonan pembukaan Rekening Efek yang diajukan oleh Nasabah dan Nasabah setuju untuk memenuhi seluruh persyaratan dan ketentuan yang ditetapkan oleh Perusahaan sekuritas.

2. Persyaratan dan Kewajiban Nasabah

2.1. Nasabah setuju untuk setiap saat mematuhi ketentuan peraturan perundang-undangan yang berlaku di Indonesia. Nasabah berjanji untuk menaati seluruh peraturan perdagangan efek yang berkaitan dengan Syarat dan Ketentuan ini, yang mengikuti dan menyesuaikan dengan hukum yang berlaku termasuk Undang-Undang Pasar Modal, peraturan-peraturan Bapepam, Bursa Efek, lembaga-lembaga penunjang, Kustodian, penjaminan dan biro administrasi efek termasuk dengan tambahan peraturan dan perubahannya bila ada.

2.2. Nasabah tidak akan menggunakan pelayanan perdagangan Efek yang diberikan oleh Perusahaan sekuritas sebagai sarana untuk melakukan tindakan yang dapat dikategorikan melanggar hukum, termasuk namun tidak terbatas kepada tindak pidana pencucian uang (*money laundering*) serta menyatakan bahwa pada saat ini Nasabah tidak sedang tersangkut perkara pidana maupun perdata di pengadilan, baik dalam jurisdiksi hukum negara Republik Indonesia maupun jurisdiksi hukum lainnya, dimana perkara tersebut bersifat material sehingga dapat mempengaruhi kemampuan Nasabah dalam memenuhi kewajiban-kewajibannya berdasarkan Syarat dan Ketentuan ini.

2.3. Nasabah berkewajiban mengisi formulir pembukaan Rekening Efek dengan benar serta melengkapi dan menyerahkan kepada Perusahaan sekuritas seluruh dokumen yang dipersyaratkan dalam formulir pembukaan Rekening Efek.

2.4. Setiap perubahan yang berhubungan dengan keterangan dalam formulir pembukaan Rekening Efek (termasuk kelengkapan dokumen yang dipersyaratkan) wajib diserahkan dan/atau diberitahukan kepada Perusahaan sekuritas selambat-lambatnya dalam waktu 1 x 24 jam setelah terjadinya perubahan. Kesalahan dalam bentuk apapun yang diakibatkan oleh tidak disampaikannya pemberitahuan mengenai perubahan keterangan mengenai Nasabah kepada Perusahaan sekuritas tidak menjadi tanggung jawab Perusahaan sekuritas.

2.5. Sebelum dilakukannya transaksi, nasabah diwajibkan memberikan deposit awal dalam bentuk dana dalam jumlah yang akan ditentukan oleh Perusahaan sekuritas.

2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori yang berkaitan dengan permasalahan yang akan dibahas sebagai pemahaman sistem yang akan dibangun.

2.2.1 Aplikasi

Aplikasi merupakan program yang berisikan perintah-perintah untuk melaksanakan pengolahan data, aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Aplikasi (*application*) juga adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word dan Microsoft Excel [3].

Dari pengertian di atas dapat disimpulkan bahwa aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Selain itu aplikasi dapat mengintegrasikan berbagai kemampuan komputer.

2.2.2 Investasi

Investasi adalah penempatan sejumlah dana dengan harapan dapat memelihara, meningkatkan nilai, atau memberikan imbal hasil (*capital gain*) yang positif. Investasi adalah penanaman modal uang dengan harapan mendapat hasil dan nilai tambah (*value added*). Investasi juga merupakan pengeluaran barang yang tidak dikonsumsi saat ini dimana berdasarkan periode waktunya, investasi terbagi menjadi tiga, yaitu investasi jangka pendek, investasi jangka menengah, dan investasi jangka panjang. Investasi merupakan komitmen sejumlah dana pada suatu periode untuk mendapatkan pendapatan yang diharapkan dimasa yang akan datang

sebagai unit kompensasi. Unit yang diinvestasikan meliputi waktu yang digunakan, tingkat inflasi yang diharapkan dan ketidakpastian masa mendatang [4].

Pada dasarnya tujuan orang melakukan investasi adalah untuk menghasilkan sejumlah uang. Secara lebih khusus ada beberapa alasan mengapa seseorang melakukan investasi antara lain [5]:

1. Untuk mendapat kehidupan yang lebih layak di masa datang.
2. Mengurangi tekanan inflasi.
3. Dorongan untuk menghemat pajak.

2.2.2.1 Bentuk Investasi

Ditinjau dari segi ruang lingkup usahanya, investasi dapat dibagi menjadi dua yakni :

a. Investasi pada aktiva nyata (*real assets* atau *real investment*), misalnya untuk pendirian pabrik-pabrik, pendirian hotel/restoran, perkebunan, dan lain-lain.
b. Investasi pada aktiva keuangan (*financial assets* atau *financial investment*), seperti pembelian surat-surat berharga, baik berupa saham maupun obligasi.

2.2.2.2 Proses Investasi

Dalam melakukan invetasi dalam sekuritas, seorang *investor* harus menentukan sekurits apa yang akan dipilih, seberapa banyak investasi tersebut akan dilakukan. Untuk mengambil keputusan tersebut diperlukan langkah-langkah sebagai berikut [5]:

1. Menentukan kebijakan investasi

Di sini pemodal perlu menentukan tujuan investasinya dan berapa banyak investasi tersebut akan dilakukan.

2. Analisis sekuritas

Dalam tahap ini, dilakukan analisis sekuritas. Pertama, ada dua filosofi dalam melakukan analisis sekuritas yaitu analisis teknikal dan fundamental. Analisis fundamental berupaya mengidentifikasi prospek perusahaan (lewat analisis terhadap faktor-faktor yang mempengaruhinya) untuk bisa memperkirakan harga saham di masa yang akan datang. Kedua, adalah mereka yang berpendapat bahwa

harga sekuritas wajar. Pada dasarnya mereka yang menganut pendapat ini berpendapat bahwa pasar modal efisien.

3. Pembentukan *portfolio*

Portfolio berarti sekumpulan investasi. Tahap ini menyangkut identifikasi sekuritas mana yang akan dipilih dan berapa proporsi dana yang akan ditanamkan pada masing-masing sekuritas tersebut.

4. Melakukan revisi *portfolio*

Tahap ini merupakan pengulangan terhadap tiga tahap sebelumnya, dengan maksud kalau perlu melakukan perubahan terhadap portfolio yang telah dimiliki.

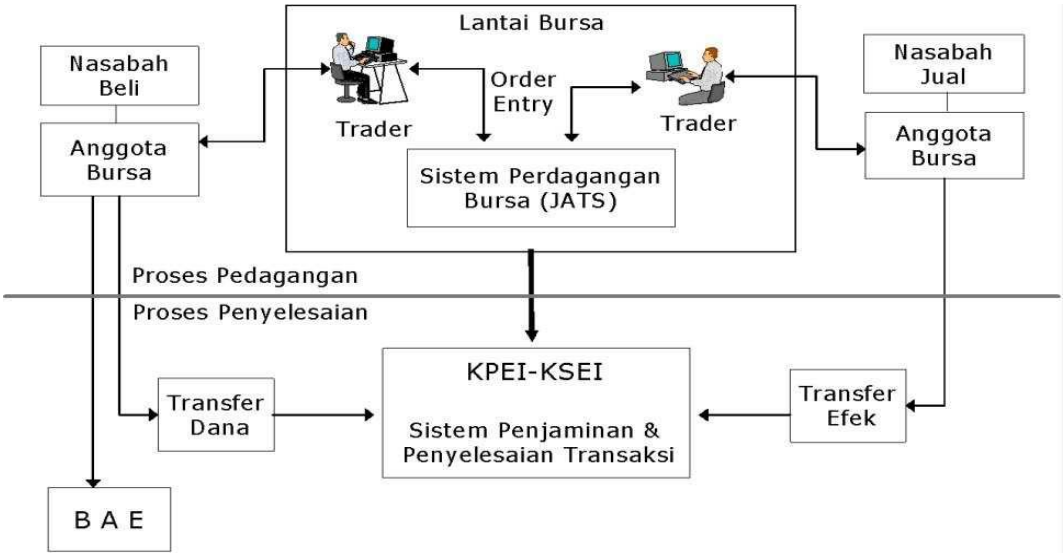
5. Evaluasi kinerja *portfolio*

Dalam tahap ini, investor melakukan penilaian terhadap kinerja portfolio, baik dalam aspek tingkat keuntungan yang diperoleh maupun risiko yang ditanggung.

2.2.3 Saham

Saham dapat didefinisikan sebagai tanda penyertaan atau pemilikan seseorang atau badan dalam suatu perusahaan atau perseroan terbatas. Wujud saham adalah, selembar kertas yang menerangkan bahwa pemilik kertas tersebut adalah pemilik perusahaan yang menerbitkan surat berharga tersebut. Porsi kepemilikan ditentukan oleh seberapa besar penyertaan yang ditanamkan di perusahaan tersebut. Saham merupakan sekuritas yang paling populer dipasar modal karena saham bisa memberikan keuntungan dalam jumlah yang besar dengan jangka waktu yang relatif singkat. Apabila perusahaan mengeluarkan satu macam saham maka saham itu disebut saham biasa (*common stock*). Apabila saham yang dikeluarkan 2 macam, yang satu adalah saham biasa dan yang lainnya adalah saham prioritas (*preferred stock*) [6].

2.2.3.1 Transaksi Saham



Gambar 2. 3 Proses Pelaksanaan Transaksi Di Bursa

Pelaksanaan Transaksi Efek di Bursa dilakukan dengan menggunakan fasilitas JATS. Transaksi Efek di Bursa hanya dapat dilakukan oleh Anggota Bursa (AB) yang juga menjadi Anggota Kliring KPEI. Anggota Bursa Efek bertanggung jawab terhadap seluruh transaksi yang dilakukan di Bursa baik untuk kepentingan sendiri maupun untuk kepentingan nasabah. Anggota Bursa Efek bertanggung jawab terhadap penyelesaian seluruh Transaksi Bursa atas nama Anggota Bursa Efek yang bersangkutan sebagaimana tercantum dalam Daftar Transaksi Bursa (DTB), termasuk Transaksi Bursa yang terjadi antara lain karena [19]:

1) kelalaian atau kesalahan PJPP dalam melaksanakan penawaran jual dan atau permintaan beli ke JATS; dan atau
2) kelalaian atau kesalahan IT *Officer Remote Trading* dalam pengoperasian Peralatan Penunjang dan atau aplikasi Anggota Bursa Efek; dan atau
3) Adanya akses yang tidak sah yang dilakukan melalui Peralatan Penunjang dan atau aplikasi Anggota Bursa Efek.

a) Segmentasi Pasar

1) Pasar Reguler : Transaksi Bursa Pasar Reguler wajib diselesaikan pada Hari Bursa ke-3 (T+3).
2) Pasar Tunai : Transaksi Bursa Pasar Tunai wajib diselesaikan pada Hari Bursa yang sama (T+0).

3) Pasar Negosiasi : Transaksi Bursa Pasar Negosiasi ditetapkan berdasarkan kesepakatan AB jual dan AB beli dan diselesaikan secara per transaksi (tidak *Netting*).

b) Biaya Transaksi

Anggota Bursa wajib membayar biaya transaksi kepada Bursa, KPEI dan KSEI yang dihitung berdasarkan nilai per transaksi Anggota Bursa. Minimum biaya transaksi yang harus dibayar AB adalah Rp 2.000.000,- (dua juta rupiah) per bulan termasuk untuk AB dalam keadaan suspensi atau SPAB-nya dibekukan. Pembayaran harus sudah efektif dalam rekening Bursa setiap bulan selambat-lambatnya pada hari kalender ke-12 bulan berikutnya.

2.2.4 Android



Gambar 2. 4 Logo Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Aliance* yang merupakan konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat mobile [7].

Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service* (GMS) dan kedua adalah yang benar – benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon selular ini diproduksi oleh HTC *Corporation* dan tersedia di pasaran pada 5 Januari 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam progam kerja Android ARM *Holdings*, Atheros *Communication*, diproduksi oleh Asustek *Computer* Inc, Garmin Ltd, Softbank, Sony Ericson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaharuan berupa perbaikan *bug* dan penambahan fitur baru.

2.2.4.1 Android SDK (*Software Development Kit*)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan.

2.2.4.2 ADT (*Android Development Tools*)

Android Develpoment Tools adalah plugin yang di desain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan

memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

2.2.4.3 Arsitektur Android

Secara garis besar Arsitektur Android dapat di jelaskan dan di gambarkan sebagai berikut [7]:

1) *Application* dan *Widgets*

Application dan *Widgets* adalah *layer* dimana *user* berhubungan dengan aplikasi saja, dimana biasanya user men-*download* aplikasi, melakukan instalasi dan menjalankan aplikasi.

2) *Application Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau member kemampuan untuk membangun aplikasi yang inovatif.

3) *Libraries*

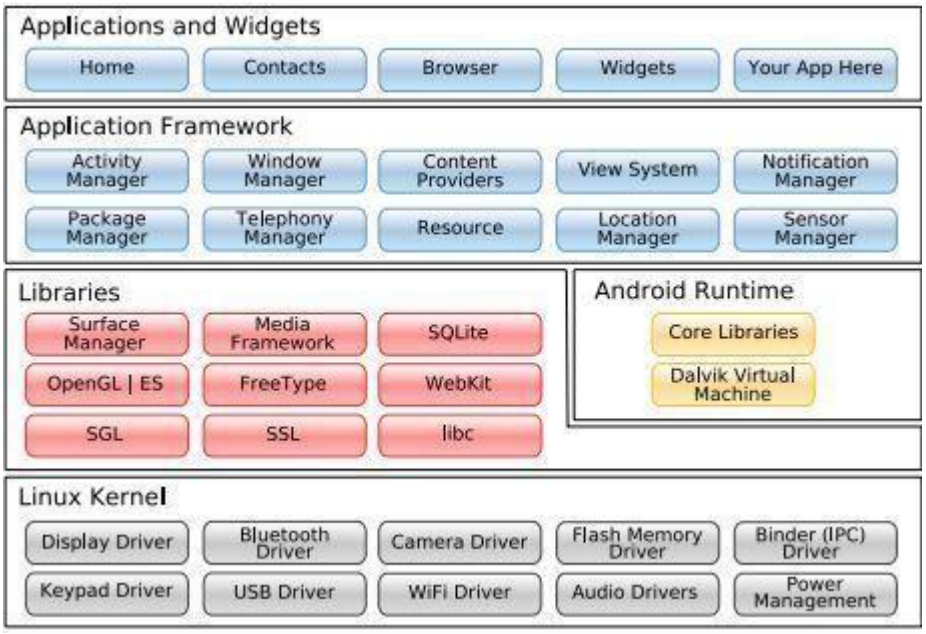
Libraries adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4) Android *Runtime*

Layer yang membuat aplikasi Android dapat djalankan dimana dalam prosesnya menggunakan Implementasi Linux.

5) Linux Kernel

Linux Kernel adalah *layer* dimana inti dari *operating system* dari Android itu berada.



Gambar 2. 5 Arsitektur Android

2.2.4.4 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan "*apt tools*" ke dalam paket Android sehingga menghasilkan file dengan ekstensi apk. Ada empat jenis komponen pada aplikasi Android yaitu :

1) *Activities*
2) *Service*
3) *Broadcast Receiver*
4) *Content Provider*

2.2.4.5 Versi Android

1. Android Versi 1.1
2. Android Versi 1.5 (*Cupcake*)
3. Android Versi 1.6 (*Donut*)
4. Android Versi 2.0/2.1 (*Éclair*)
5. Android Versi 2.2 (*Froyo*)

6. Android Versi 2.3 (*Gingerbread*)
7. Android Versi 3.0 (*Honeycomb*)
8. Android Versi 4.0 (*Ice Cream Sandwich*)
9. Android Versi 4.1 (*Jelly Bean*)
10. Android Versi 4.4 (*KitKat*)
11. Android Versi 5.0 (*Lollipop*)
12. Android Versi 6.0 (*Marshmallow*)

2.2.5 Android Studio



Gambar 2. 6 Logo Android Studio

2.2.5.1 Definisi Android Studio

Android Studio adalah IDE resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Di atas IntelliJ yang kuat *code editor* dan pengembang alat, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas Anda ketika membangun aplikasi Android, seperti [8]:

1. Sebuah fleksibel berbasis Gradle membangun sistem
2. Membangun varian dan beberapa generasi file APK
3. Kode template untuk membantu Anda membangun fitur aplikasi umum
4. Sebuah *layout editor* kaya dengan dukungan untuk drag dan drop tema *editing*
5. alat *Lint* untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya

6. Kode menyusut dengan ProGuard dan sumber daya menyusut dengan Gradle
7. *Built-in* dukungan untuk *Google Cloud Platform* , sehingga mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine.*

2.2.5.2 Struktur Proyek

Setiap proyek di Android Studio berisi satu atau lebih modul dengan file kode sumber dan file sumber daya. Berbagai jenis modul meliputi :

1) Android app modules
2) Test modules
3) Library modules
4) App Engine modules

Secara default, Android Studio menampilkan file proyek Anda dalam tampilan proyek Android. Pandangan ini diselenggarakan oleh modul untuk menyediakan akses cepat ke file kunci sumber proyek Anda. Semua file membangun terlihat di tingkat atas di bawah *Script Gradle* dan setiap modul aplikasi mengandung tiga unsur berikut :

1) Manifests : file Manifest.
2) Java : Sumber file kode.
3) Res : file Sumber Daya.

Struktur proyek Android pada disk berbeda dari representasi pipih ini. Untuk melihat struktur file yang sebenarnya dari proyek, pilih Proyek dari Proyek drop-down. Anda juga dapat menyesuaikan tampilan dari file proyek untuk fokus pada aspek-aspek tertentu dari pengembangan aplikasi Anda. Misalnya, memilih Masalah tampilan proyek Anda menampilkan link ke file sumber yang berisi setiap diakui coding dan sintaksis kesalahan, seperti hilang elemen XML tag penutup dalam *file layout.*

2.2.5.3 Versi Android Studio

1) Android Studio v0.1.x (Mei 2013)
2) Android Studio v0.2.x (Juli 2013)

3) Android Studio v0.3.2 (Oktober 2013)
4) Android Studio v0.4.2 (Januari 2014)
5) Android Studio v0.4.6 (Maret 2014)
6) Android Studio v0.5.2 (Mei 2014)
7) Android Studio v0.8.0 (Juni 2014)
8) Android Studio v0.8.6 (Agustus 2014)
9) Android Studio v0.8.14 (Oktober 2014)
10) Android Studio v1.0 (Desember 2014)
11) Android Studio v1.0.1 (Desember 2014)
12) Android Studio v1.1.0 (Februari 2015)
13) Android Studio v1.2.0 (April 2015)
14) Android Studio v1.2.1 (Mei 2015)
15) Android Studio v1.2.2 (Juni 2015)
16) Android Studio v1.3.0 (Juli 2015)
17) Android Studio v1.3.1 (Agustus 2015)
18) Android Studio v1.3.2 (Agustus 2015)
19) Android Studio v1.4.0 (September 2015)
20) Android Studio v1.4.1 (Oktober 2015)
21) Android Studio v1.5.0 (November 2015)
22) Android Studio v1.5.1 (Desember 2015)
23) Android Studio v2.0.0 (April 2016)
24) Android Studio v2.1.0 (April 2016)
25) Android Studio v2.1.1 (Mei 2016)

2.2.6 Java



Gambar 2. 7 Logo Java

2.2.6.1 Definisi Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web* [9].

2.2.6.2 Sejarah perkembangan Java

Bahasa pemrograman Java terlahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick

Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program *Java Oak* pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.

Perusahaan baru ini bertambah maju, jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi *release* Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil

dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah *Java*).

2.2.6.3 Versi Java

Versi awal Java ditahun 1996 sudah merupakan versi *release* sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya :

1. java.lang : Peruntukan kelas elemen-elemen dasar.
2. java.io : Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.
3. java.util : Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. java.net : Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. java.awt : Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
6. java.applet : Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah *web*.

2.2.6.4 Kelebihan Java

1. ***Multiplatform***. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*/sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja*. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/*bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan.
2. **OOP** (*Object Oriented Programming* - Pemrogram Berorientasi Objek)
3. **Perpustakaan Kelas Yang Lengkap**, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus

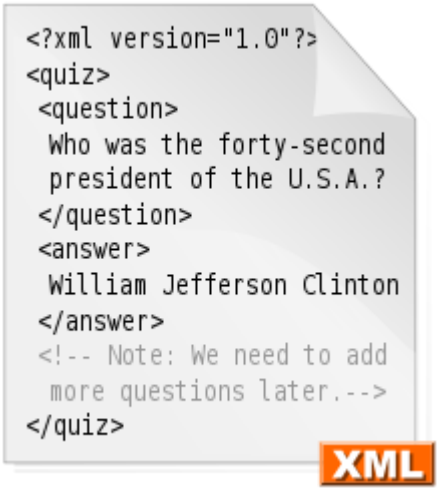
membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

4. **Bergaya C++**, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. **Pengumpulan Sampah Otomatis**, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2.6.5 Kekurangan Java

1. **Tulis sekali, jalankan di mana saja** - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
2. **Mudah didekompilasi**. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-*reverse-engineer*.
3. **Penggunaan memori yang banyak**. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan *Object* Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.2.7 XML



Gambar 2. 8 Contoh Dokumen XML

2.2.7.1 Definisi XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet [10].

2.2.7.2 Pengenalan XML

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar.

2.2.7.3 Tipe XML

XML memiliki tiga tipe file :

1. XML, merupakan standar format dari struktur berkas (*file*).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.

3. XSD, merupakan standar yang mendefinisikan struktur *database* dalam XML.

2.2.8 PHP (*Hypertext Preprocessor*)

PHP adalah sebuah bahasa pemograman yang berjalan dalam sebuah *web server (server side*). PHP adalah bahasa skrip yang sangat populer digunakan untuk mengembangkan aplikasi-aplikasi *web*, meskipun sebenarnya PHP juga dapat digunakan untuk mengembangkan aplikasi desktop, baik *console* maupun yang berbasis GUI [11].

2.2.8.1 Konsep Dasar PHP

Kode PHP diawali dengan tanda lebih kecil (<) dan diakhiri dengan tanda lebih besar (>). Ada empat cara untuk menuliskan skrip PHP, yaitu :

1. <? Hallo ?>
2. <? php Hallo ?>
3. <% Hallo %>
4. <SCRIPT LANGUAGE="php">Hallo</script>

Pemisah antara instruksi adalah titik koma (;) dan untuk membuat atau menambahkan komentar/standar penulisan adalah : /* komentar */, // komentar, # komentar. Cara penulisan dibedakan menjadi *Embedded* dan *Non Embedded script*.

2.2.8.2 Variabel PHP

Variabel PHP digunakan untuk menyimpan data yang nilainya dapat berubah-ubah. Dalam bahasa PHP, variabel dimulai dengan tanda "$". Aturan penulisan variabel antara lain sebagai berikut :

a. Hanya ada 3 karakter yang dapat digunakan untuk nama variable, yaitu huruf, angka dan garis bawah.
b. Karakter pertama setelah tanda "$" harus huruf atau garis bawah.
c. Jika nama variable lebih dari satu kata, tidak boleh ada tanda spasi diantara keduanya.

2.2.8.3 Kelebihan PHP

PHP *Hypertext Preprocessor* adalah skrip yang berjalan dalam *server side* yang ditambahkan dalam HTML. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman HTML tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* ini membuat pekerjaan skrip tersebut dikerjakan di *server* sedangkan yang dikirimkan ke *browser* adalah hasil proses dari skrip tersebut yang sudah berbentuk HTML. Keunggulan dari sifatnya yang *server side* tersebut adalah :

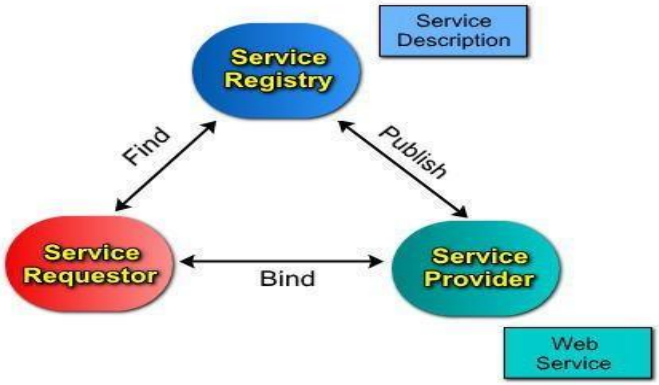
a. Tidak diperlukan adanya kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena *server*-lah yangakan mengerjakan skrip tersebut. Hasil yang dikirimkan kembali ke *browser* biasanya dalam bentuk teks ataupun gambar sehingga dapat dikenali oleh *browser* apapun.
b. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, contohnya : hubungan kedalam *database*.
c. Skrip asli tidak terlihat sehingga keamanam lebih terjamin.

2.2.9 *Web Service*

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Sistem-sistem lainnya berinteraksi dengan *web service* menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML [12].

Definisi diatas diberikan oleh *World Wide Web Consortium*(W3C) yang merupakan badan yang menciptakan dan mengembangkan standar *web service.* Tetapi secara umum, *web service* tidak terbatas hanya pada standar SOAP saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui *internet* menggunakan protokol standar *internet* dan menggunakan XML sebagai format pesannya.

2.2.9.1 Arsitektur *Web service*



Gambar 2. 9 Arsitektur *Web Service*

Pada gambar diatas, ada tiga komponen yang membuat *web service* berjalan. Ketiga komponen itu adalah [12]:

1. ***Service provide****r*, merupakan pemilik *Web service* yang berfungsi menyediakan kumpulan operasi dari *Web service*.
2. ***Service requesto****r*, merupakan aplikasi yang bertindak sebagai klien dari *Web service* yang mencari dan memulai interaksi terhadap layanan yang disediakan.
3. ***Service registry***, merupakan tempat dimana *Service provider* mempublikasikan layanannya. Pada arsitektur *Web service*, Service registry bersifat optional. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis *software* yang memiliki *platform* dan sistem operasi yang berbeda.

2.2.9.2 Jenis-Jenis *Web service*

Jenis *web service* dapat dibagi menjadi dua, yaitu REST dan SOAP.

1. *Representational State Transfer* (REST)

REST adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa

digunakan adalah fungsi *GET, POST, PUT* atau *DELETE*. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi
Client [13].

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi pada *resource*nya itu, sedangkan aplikasi lowongan pekerjaan membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource* data lowongan. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *web service* yang menggunakan REST adalah: Flickr API(*Application Program Interface*), YouTube API, Amazon API.

2. *Simple Object Access Protokol* (SOAP)

SOAP merupakan protokol standar yang ringan dan ditujukan untuk pertukaran informasi dalam desentralisasi sistem. Protokol ini melakukan pemanggilan metode dalam bahasa XML (*Extensible Markup Language*). Namun, selain itu juga kita dapat menggunakan format pertukaran data lainnya yaitu JSON (*JavaScript Object Notation*). Dengan demikian, SOAP adalah suatu mekanisme yang sederhana untuk melakukan pertukaran struktur dan tipe informasi dalam lingkungan yang tersebar dan terdistribusi menggunakan XML/JSON. Dokumen SOAP yang digunakan untuk melakukan *request* disebut SOAP *request* sedangkan dokumen SOAP yang diperoleh dari web service disebut dengan SOAP *response* [13].

2.2.9.3 Web service Definition Language (WSDL)

WSDL merupakan suatu dokumen berbasis XML yang berfungsi untuk menjelaskan fungsi *public* dan metode-metode yang tersedia dalam *web service*. Selain itu, WSDL juga mengatur *end-point* pada pesan-pesan yang berisi dokumen

maupun informasi prosedur. Dokumen WSDL sanagat kompleks, dan memungkinkan untuk mendeskripsikan *end-point* serta pesan yang ada pada protokol jaringan. Salah satu bagian dokumen WSDL yang cukup penting dalam melakukan interaksi dengan SOAP, HTTP, serta MIME adalah *binding*. SOAP dapat berjalan sendiri tanpa bantuan WSDL, tetapi hal ini akan menyulitkan pengembang di sisi *client* karena tidak tahu harus mengharapkan operasi dan bentuk data apa. Karena itu WSDL diciptakan sebagai pendefinisi sebuah *web service*. Dalam tugas akhir ini, *web service* yang dibangun akan dijabarkan menggunakan WSDL [12].

2.2.9.4 UDDI (Uniform Description, Discovery, and Integration)

Dalam penerapannya, *web service* memerlukan sumber dari semua informasi tentang *SOAP message*, cara membuat respons, mengumumkan *service* yang ada, dan sebagainya. Dengan demikian, aplikasi dapat menggunakannya untuk mencari WSDL. Untuk memenuhi hal tersebut, diperlukan UDDI yang menyediakan *service* untuk mencari informasi yang diperlukan.

UDDI memungkinkan suatu perusahaan untuk mengumumkan detail dari *web service* yang tersedia, biasanya melalui kata kunci atau deskripsi. Dengan adanya UDDI tersebut, kita dapat mengetahui (*discover*) dimana *web service* perusahaan berada [12].

2.2.9.5 JSON (JavaScript Object Notation)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. Kelebihan-kelebihan dari JSON adalah [14]:

1. Format Penulisan : Untuk merepresentasikan sebuah struktur data yang rumit dan berbentuk hirarkis penulisan JSON relatif lebih terstruktur dan mudah.

2. Ukuran : Ukuran karakter yang dibutuhkan JSON lebih kecil dibandingkan XML untuk data yang sama.
3. *Browser Parsing* : Proses parsing merupakan proses pengenalan token atau bagian-bagian kecil dalam rangkaian dokumen XML/JSON.

JSON terbuat dari dua struktur :

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

2.2.10 *Web Mining*

Web mining merupakan satu bidang data mining yang memusatkan pada proses *mining* data pada *World Wide Web* (WWW). Pusat informasi ini tentu akan menjadi sumber pengetahuan yang menarik dimana *data mining* dapat diterapkan untuk mendapatkan berbagai pengetahuan yang memiliki nilai manfaat yang tinggi. Berdasarkan tipe data mentah yang diolahnya. Target data dari *Web mining adalah* menganalisis data dari web, seperti data akses pengunjung, struktur halaman *web, format* halaman *web* dan sebagainya. Berdasarkan target analisisnya, *web mining* dibagi menjadi tiga kategori yaitu [15]:

1. *Web structure mining*

Proses yang menggunakan teori *graph* untuk menganalisis simpul (*node*) dan keterhubungan struktur dari situs. Menurut tipe dari struktur web, *web structure mining* terbagi menjadi 2 (dua). Pertama adalah mengekstrak dari pola *hyperlink* di web. Kedua dari *web structure mining* adalah mining terhadap struktur dokumen. Yang dimaksud sebagai struktur dokumen adalah menganalisa struktur dari bahasa yang digunakan dalam web, yaitu bahasa HTML (*Hyper Text Markup Language*), atau XML (*eXtensibel Markup Language*) di dalam halaman.

2. *Web Content Mining*

Proses untuk mendapatkan informasi yang berguna dari isi (*content*) di *web*. Isi (*content*) dapat berupa *text*, *image*, *audio* dan *video*. *Web content mining* terkadang disebut sebagai *web text mining*, karena teks merupakan bagian dari *web* yang paling banyak tersedia.

3. *Web Usage Mining*

Teknik *data mining* yang berusaha mengungkap pola penggunaan dari halaman web, dalam rangka coba untuk memahami dan meningkatkan pelayanan kebutuhan dari aplikasi berbasis *web*. Pada jenis *structure* dan *content mining*, yang dianalisa atau digali adalah data didalam *web* itu sendiri, namun pada *web usage mining* yang dianalisa adalah pengguna atau pengunjung dari halaman *web*.

2.2.10.1 Perbedaan Jenis-jenis *Web Mining*

Tabel 2. 1 Perbedaan Jenis-jenis *Web Mining*

Topic	***Web Mining***			
	Web structure mining		***Web content mining***	***Web usage mining***
	IR View	***DB View***		
View of Data	*Unstructured, Semistructured*	*Semistructured, Web as DB*	*Links structure*	*Interactivity*
Main Data	*Text documnts Hypertext docs*	*Hypertext docs*	*Links structure*	*-Server logs, -Browser logs*
Representation	*-Bag of words, n-grams -Terms, phrase -Concept or ontology -Relational*	*OEM,Relational*	*Graph*	*-Relational table, -Graph*
Method	*-TFIDF and variants,*	*Propietary algorithms,*	*Propietary algorithms*	*-Machine Learning, -Statistical,*

	-Machine Learning, -Statistic (NLP)	*ILP, Association rules*		*-(Modified)*
				association rules
Application Categories	*Categorization, Clustering, Finding extraction rules, Finding pattern in text, User modelling*	*Finding frequent substructure, Website schema discovery*	*Categorization , Clustering*	*-Site construction, addaption and management -Marketing, -User modelling*

2.2.11 MYSQL

MySQL merupakan *software database* yang paling populer di kalangan para pengembang aplikasi *database*. MySQL juga adalah suatu perangkat lunak yang menganut atau mengimplementasikan model basis data relasional maka MySQL disebut sebagai *Relational Database Management System* (RDBMS). MySQL merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*) [11].

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

baik. Beberapa contoh aplikasi yang menggunakan MySQL adalah Joomla, Wordpress, MyBB, phpBB, dan masih banyak lagi.

2.2.11.1 Kelebihan MySQL

Beberapa kelebihan yang menjadikan para pengembang aplikasi *database* memilih untuk menggunakan MySQL dibanding dengan RDBMS yang lain. Banyaknya pengguna MySQL yang kian meningkat secara drastis ini mengakibatkan nama MySQL menjadi melambung tinggi dan menjadikannya se-populer seperti saat sekarang. Berikut ini beberapa keuntungan dari MySQL [11]:

1. Fleksibel

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

2. Performa Tinggi

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

3. Lintas *Platform*

   MySQL dapat digunakan pada *platform* atau lingkungan (dalam hal ini sistem operasi) yang beragam, bisa Microsoft Windows, Linux, atau UNIX. Ini menyebabkan proses migrasi data (bila dibutuhkan) antar sistem operasi dapat dilakukan secara lebih mudah. Misal, jika kita ingin mengganti sistem operasi pada mesin *server*.

4. Gratis

   MySQL dapat digunakan secara gratis. Meskipun demikian, ada juga *software* MySQL yang bersifat komersil. Biasanya yang sudah ditambahi dengan kemampuan spesifik dan mendapat pelayanan dari *technical support.*

5. Proteksi Data yang Andal

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

6. Komunitas Luas

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

a. Bahasa MySQL

Seperti halnya SQL *engine* yang lain, Perintah dalam SQL diklasifikasikan menjadi lima bagian besar, yaitu [11]:

a. *Data Defenition Language* (DDL), Kumpulan perintah yang berfungsi untuk mendefinisikan *database,* struktur tabel, indeks, dan objek-objek lain dalam *database*. Perintah-perintah yang digolongkan dalam DDL adalah *create, alter* dan *drop*.
b. *Data Manipulation Language* (DML), Kumpulan perintah yang digunakan untuk melakukan manipulasi data di dalam tabel. Perintah-perintah yang digolongkan dalam DML adalah *update, insert,* dan *delete*.
c. *Data Control Language* (DCL), Kumpulan perintah yang digunakan untuk mengontrol (memberi/mencabut) hak akses ke/dari seorang *user*. Dua perintah utama dalam DCL adalah grant dan *revoke*. *Grant* digunakan untuk memberikan hak akses kepada *user* tertentu, sedangkan *revoke* mencabut hak akses dari *user* tertentu dalam *database*.

d. *Transactional Control Language* (TCL), Kumpulan perintah yang digunakan untuk mengontrol (menyimpan/membatalkan) transaksi data. Perintah-perintah yang digolongkan dalam TCL adalah *commit* dan *rollback. Commit* digunakan untuk menyimpan perubahan-perubahan yang dilakukan terhadap *database* secara permanen, sedangkan *rollback* digunakan untuk membatalkan transaksi atau perubahan-perubahan yang telah dilakukan ke dalam *database*.

e. *Data Query Language* (DQL), Kumpulan perintah yang digunakan untuk melakukan seleksi data.

b. Melihat *Database*

Kita bisa melihat daftar *database* yang ada di MySQL dengan mengetikkan baris perintah berikut : **mysql> SHOW DATABASES;**

MySQL akan menunjukkan *list*/daftar dari database pada *server*. Jika ini adalah sebuah *server* baru (sebagai contoh : jika kamu menginstal *server* ini sendiri) daftarnya harus kelihatan seperti ini :

```
----------------------
| Database |
----------------------
| mysql |
| test |
----------------------


2 rows inset (0.11sec)
```

c. Membuat *Database*

Berikut ini adalah cara membuat sebuah *database* pada MySQL. Sebagai latihan kita membuat *database* dengan nama “Saham_db”, dan perintahnya adalah sebagai berikut :

mysql> create database Saham_db;

Query OK,1 row affected (0.03sec)

d. Menghapus *Database*

Untuk menghapus *database* dapat dilakukan dengan men"*dropping*"nya, dan perintahnya adalah :

mysql> drop database Saham_db;

2.2.12 *Object Oriented Programming*

Secara spesifik, pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme) dan *inheritance* (pewarisan) [16].

2.2.12.1 Konsep Dasar Berorientasi *Object*

Pengembangan sistem berorientasi objek berbeda dari pengembangan konvensional yang memandang perangkat lunak sebagai fungsi dan data yang terisolasi. Program perangkat lunak adalah sekumpulan mekanisme yang melakukan aksi-aksi tertentu pada data tertentu. Pada pendekatan konvensional, kebanyaan berfous pada fungs. Namun juga terdapat pendekatan yang fokus pada data terutama pada kubu basis data dan pemodelan informasi. Sementara itu, pandangan berorientasi objek berpusat pada objek yang mengkombinasikan data dan fungsionalitas. Berikut ini adalah beberapa kriteria-kriteria penting pendekatan objek yang harus dipahami tentang metodologi berorientasi objek [16]:

a. *Seamlessness*

   Konsep berorientasi objek dapat diterapkan di seluruh siklus hidup pengembang sistem (*System Developmnent life cycle*) dari analisis, perancangan, sampai implementasi.

b. Kelas

   Pendekatan berorientasi objek berdasar pada pentingnya kelas. Kelas adalah elemen perangkat lunak yang mendeskripsikan tipe data abstrak (ADT –

Abstract data type) dimana implementasinya dapat secara parsial atau total. Implementasi parsial bila di kelas terdapat operas/metode abstrak.

c. *Assertion*

Fitur tipe data abstrak harus secara formal menspesifikasikan properti kelas Assertion berperan pada prakondisi (*precondition*), poskondisi (*postcondition*), invarian kelas. Assertion mendeskripsikan efek fitur-fitur di objek independen terhadap implementasi fitur-fitur itu.

d. Kelas Merupakan Cara Pemodulan

Pendekatan berorientasi objek merupakan teknik pembentukan arsitektur perangkat lunak. Dampak utama penerapannya adalah pada struktur modul dari sistem perangkat lunak. Peran utamanya di sistem berorientasi objek dimainkan oleh kelas. Pada pendekatan murni berorientasi objek maka kelas merupakan satu-satunya pemodulan.

e. Kelas merupakan *type*

Kelas cukup ampuh untuk menghilangkan kebutuhan mekanisme *typing* yang lain. Setiap *type* harus didasarkan pada kelas.

f. *Exception handling*

Sistem perangkat lunak seharusnya mampu menangani kondisi abnormal secara terstruktur dengan baik.

g. *Genericity*

Agar penerapan *typing* menjadi praktis adalah kemampuan untuk mendefinisikan *type-parameterized classes* atau disebut generik.

h. Pewarisan

Pendekatan berorientasi objek harus memungkinkan mendefinisikan kelas yang merupakan turunan kelas yang lain. Pewarisan dapat berupa pewarisan tunggal (*single inheritance*) dan pewarisan jamak (*multiple inheritance*).

i. *Redefinition*

Kita mungkin perlu mengubah implementasi atau properti lain dari fitur-fitur yang diwariskan. Redefinisi dapat mempengaruhi fitur, penanda (tipe argumen dan hasil) dan spesifikasi.

j. *Polymorphism*

Dimungkinkan untuk entitas ditautkan ke objek-objek beragam tipe. *Polymorphism* dikendalikan oleh pewarisan.

k. *Dynamic binding*

Dynamic binding mempunyai pengaruh utama pada aplikasi berorientasi objek. Mekanisme ini memungkinkan pengembang menulis panggilan secara sederhana untuk beberapa panggilan yang dapat dimungkinkan, bergantung pada situasi saat sebenarnnya panggilan itu dieksekusi.

l. *Run-time type interogation*

Perangkat lunak memerlukan mekanisme untuk mengakses objek dengan cara yang aman tanpa melanggar konstrain *static typing*. Mekanisme ini harus dirancang agar tidak emniadakan manfaat-manfaat dari *dynamic binding* dan *polymorphism*.

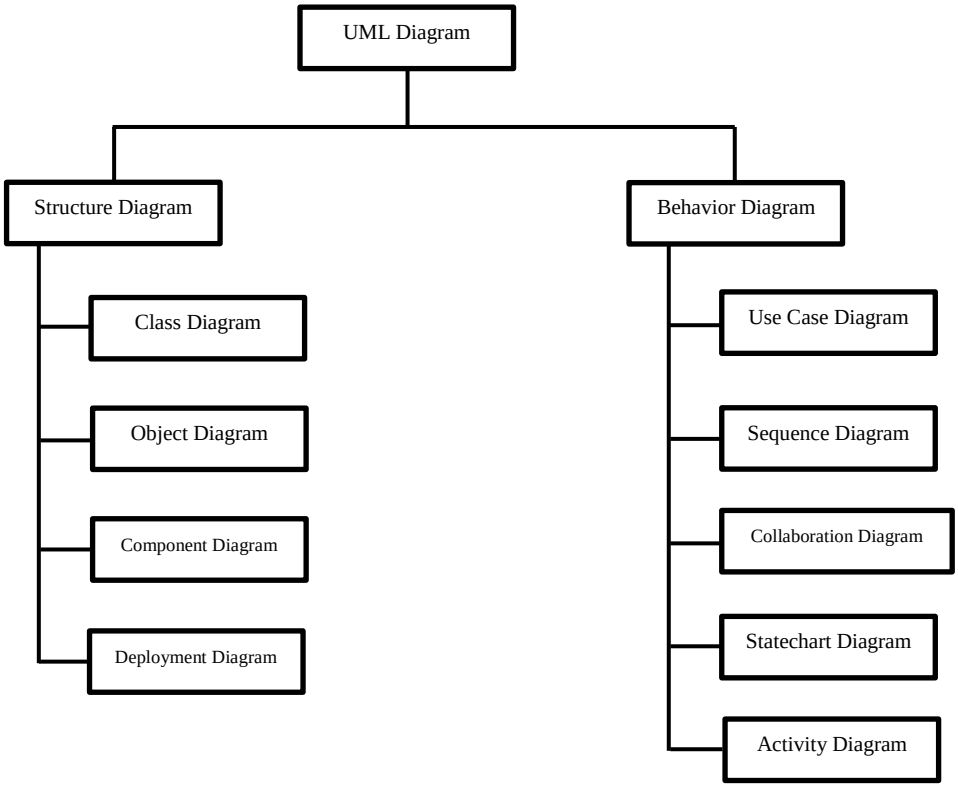
2.2.12.2 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMG (*Object Management Group*) [16].

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya yaitu : Grady Booch OOD (*Object Oriented Design*), James Rumbaugh OMT (*Object Modeling Technique*) dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*).

2.2.12.3 UML *Diagram*

Pada UML terdiri dari 13 macam diagram yang dikelompokan dalam 2 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.10.



Gambar 2. 10 UML *Diagram*

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

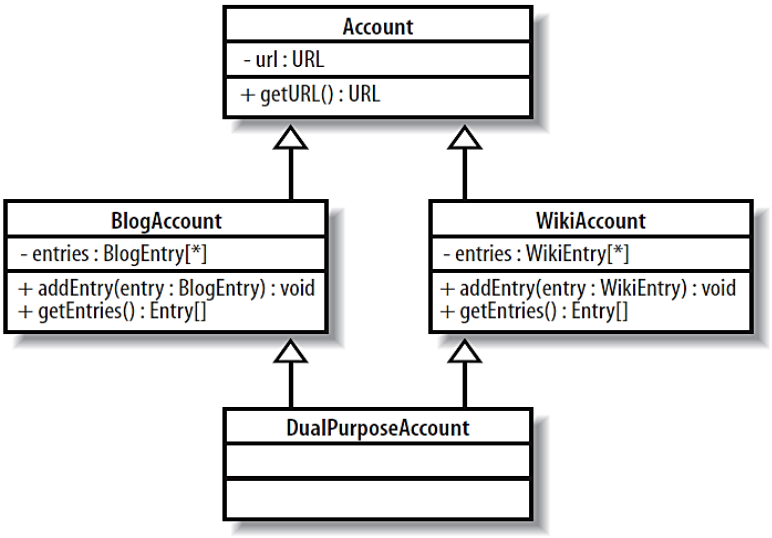
1. *Structure diagrams* yaitu diagram untuk memvisualisasikan, mendokumentasi, menspesifikasikan, dan membangun aspek statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu diagram untuk memvisualisasikan, mendokumentasi, menspesifikasikan, dan membangun aspek dinamis dari sistem yang dimodelkan.

2.2.12.4 *Class Diagram*

Diagram Kelas menunjukkan sekumpulan kelas, interface dan kolaborasi dan keterhubungan. Diagram kelas ditujukan untuk pandangan statik terhadap sistem. Kelas memiliki apa yang disebut nama, atribut, metode atau operasi [16].

1. Nama kelas harus unik. Nama akan menjadi identifier di program, seharusnya sedini mungkin dipilih nama yang memenuhi aturan (semua) bahasa pemrograman.
2. Atribut adalah properti bernama di kelas yang mendeskripsikan *range* nilai yang dipunyai instan kelas. Kelas dapat mempunyai sejumlah atribut atau tidak sama sekali.
3. Operasi atau metode adalah implementasi layanan yang dapat diminta pada sembarang objek kelas untuk mempengaruhi perilaku sistem. Kelas dapat mempunyai sejumlah operasi atau tidak sama sekali.

Berikut Contoh *Class Diagram* dapat dilihat pada Gambar 2.11 [17].



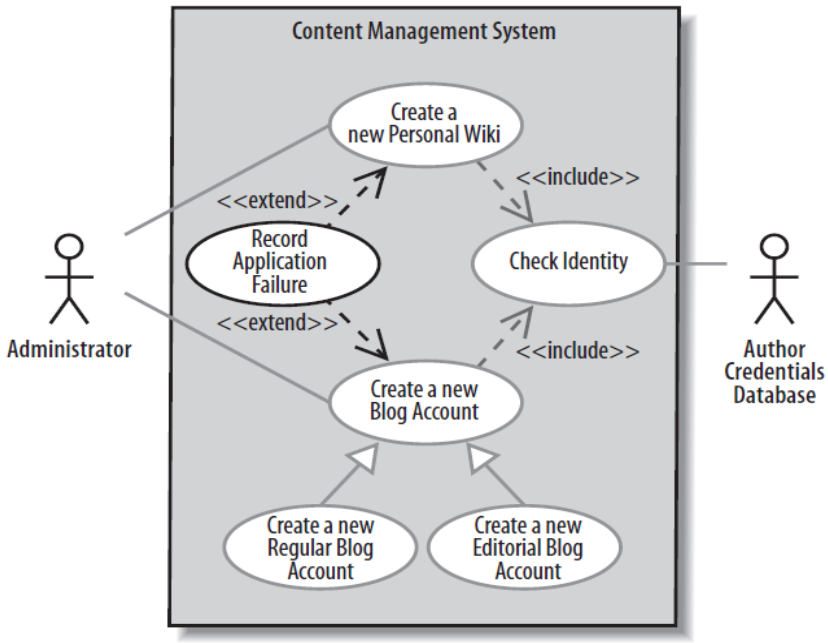
Gambar 2. 11 Contoh *Class Diagram*

2.2.12.5 *Use Case Diagram*

Use case Diagram merupakan salah satu diagram untuk memodelkan aspek perilakau sistem. Masing-masing diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. Diagram *use case* adalah penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram-diagram *use case* merupakan pusat pemodelan perilaku sistem, subsistem, dan kelas. Diagram *use case* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Ada empat hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* [16].

1. Sistem yaitu sesuatu yang hendak kita bangun.
2. Relasi adalah relasi antara aktor dengan *use case*
3. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
4. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.12 [17].



Gambar 2. 12 Contoh *Use Case Diagram*

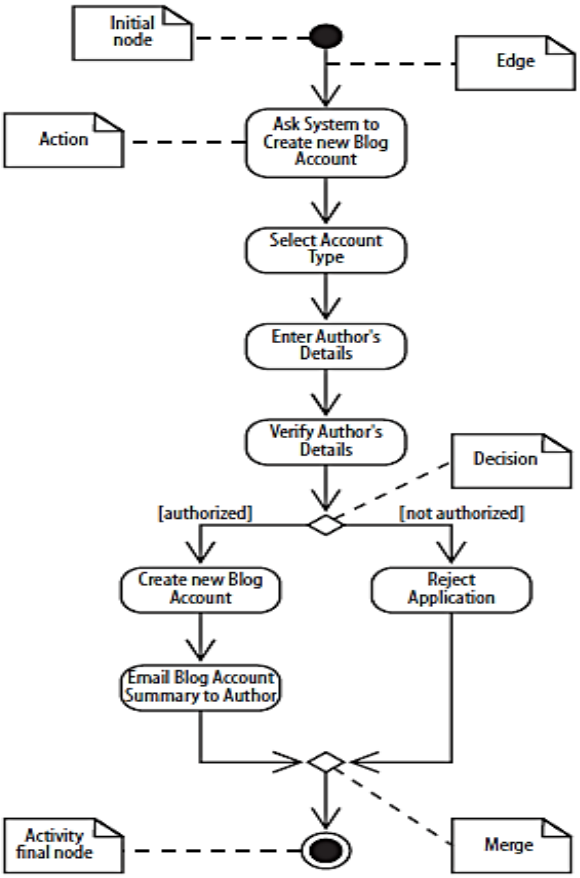
2.2.12.6 *Activity Diagram*

Diagram aktivitas adalah diagram flowchart yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain di sistem. Diagram aktivitas ini digunakan untuk memodelkan aspek dinamis sistem. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut [16]:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokkan 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.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut Contoh *Activity Diagram* dapat dilihat pada Gambar 2.13 [17].



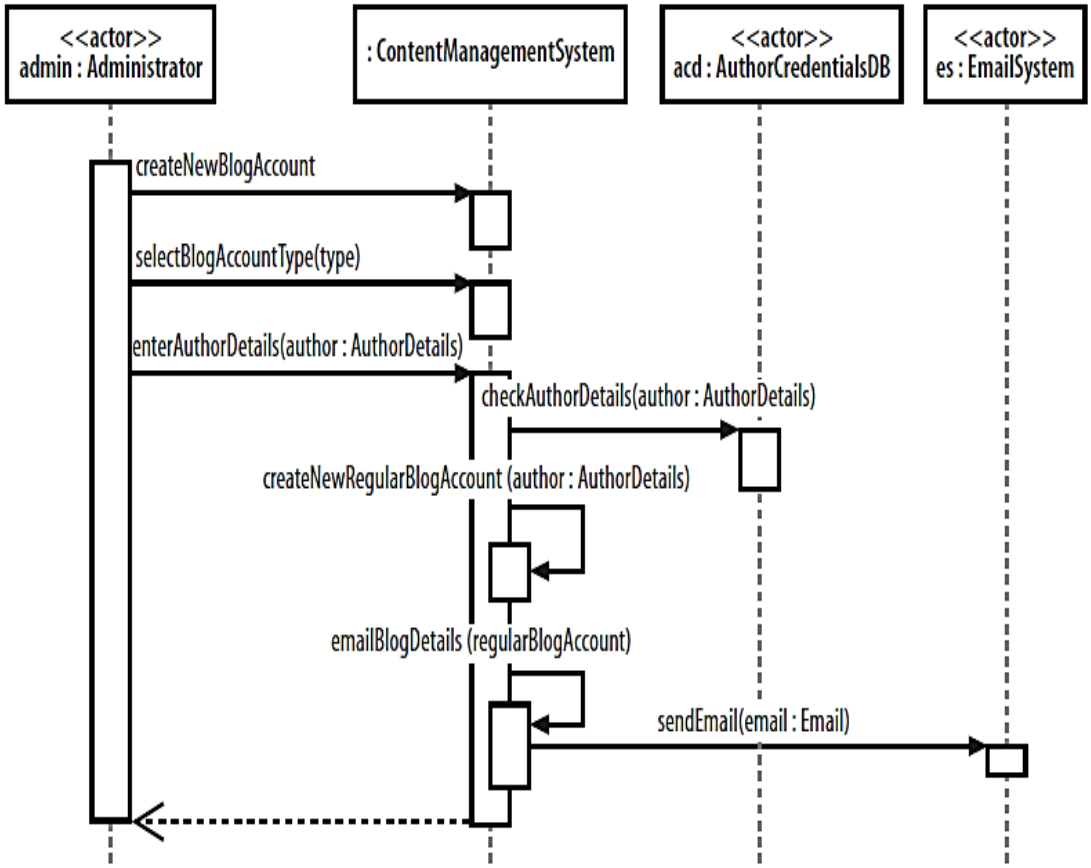
Gambar 2. 13 Contoh *Activity Diagram*

2.2.12.7 *Sequence Diagram*

Diagram sekuen menunjukan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada sisi basis keberurutan waktu dari pesan-pesan yang terjadi. Diagram sekuen menunjukan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukan barisan kejadian, bukan

perwaktuan nyata. Kecuali untuk sistem waktu nyata yang mengharuskan konstrain barisan terjadi [16].

Berikut Contoh *sequence diagram* dapat dilihat pada gambar 2.14 [17].



Gambar 2. 14 Contoh *Sequence Diagram*

2.2.13 Metode Pengujian Sistem

Metode pengujian sistem terdiri dari Pengujian *black box* dan *beta* yang dilakukan untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan.

2.2.13.1 Pengujian *Black Box*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianalogikan seperti melihat suatu kotak hitam yang hanya bisa dilihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. *Black box* testing melakukan evaluasi untuk menemukan kesalahan dalam kategori berikut[20]:

a. Fungsi tidak benar atau hilang.
b. Kesalahan *interface* atau antarmuka.
c. Kesalahan dalam struktur data atau akses *database eksternal*.
d. Kesalahan kinerja atau perilaku dan kesalahan inisialisasi dan terminasi.

2.2.13.2 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai tanggapan siswa terhadap aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari siswa.

2.2.13.3 Skala Likert

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala Likert. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena [21].

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan negative yang berfungsi untuk mengukur sikap negative objek. Skor pernyataan positif dimulai dari 1 untuk sangat tidak setuju (TS), 2 untuk tidak setuju (KS), 3 untuk raguragu (CS), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (CS), 4 untuk tidak setuju (KS), dan 5 untuk sangat tidak setuju (TS). Beberapa menghilangkan option "Ragu-ragu" dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. Skala Likert digunakan untuk mengukur kesetujuan dan ketidaksetujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Kekurangan Skala Likert :

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai error dari respons yang terjadi.

Kelebihan Skala Likert :

1. Mudah dibuat dan di terapkan.
2. Skala Likert lebih mudah membuatnya dibanding lain sepertti skala Thurstone.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.
4. Jawaban suatu item dapat berupa alternative, sehingga informasi mengenai item tersebut diperjelas.
5. Reliabilitas pengukuran bisa diperoleh dengan jumlah item tersebut diperjelas.
6. Karena jangka responsi yang lebih besar membuat skala Likert dapat memberikan keterangan yang lebih jelas dan nyata tentang pendapatan atau sikap responden tentang isu yang dipertanyakan.

2.2.13.4 Kuesioner

Menurut Depdikbud kuesioner atau angket adalah suatu alat pengumpul data yang berupa serangkaian pertanyaan yang diajukan pada *responden* untuk mendapat jawaban. Angket merupakan teknik pengumpulan data yang dilakukan dengan mengadakan komunikasi dengan sumber data. Menurut KBBI (Kamus Besar Bahasa Indonesia) kuesioner merupakan alat riset atau survei yang terdiri atas serangkaian pertanyaan tertulis, bertujuan mendapatkan tanggapan dari kelompok orang terpilih melalui wawancara pribadi atau melalui pos daftar pertanyaan.

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Selain itu, kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar di wilayah yang luas. kuesioner dapat berupa pertanyaan/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 pengiriman kuesioner kepada responden tidak perlu melalui pos [22].