

BAB II

LANDASAN TEORI

2.1.Konsep Dasar Sistem

Menurut Furqon dalam Wiwid (2016:2), “Sistem informasi penjualan merupakan suatu sistem yang berfungsi untuk mengolah data-data terkait dengan kegiatan penjualan baik dari transaksi pembelian sampai transaksi penjualan digunakan untuk mendukung kegiatan penjualan tersebut.”

Sistem informasi penjualan merupakan salah satu dari sistem informasi yang terpenting pada perusahaan umumnya. Sistem informasi penjualan ini bertujuan untuk membantu manajer dalam berbagai hal seperti :

1. Membantu manajemen dalam pengambilan keputusan.
2. Manajemen dapat menerima laporan lebih sering dan terperinci.
3. Manajemen dapat memonitor prestasi produk, pasar, karyawan, penjualan dan berbagai unit pemasaran lainnya.

Sistem informasi penjualan ini sangat berperan dalam setiap perusahaan, agar aktivitas penjualan yang dilakukan dapat cepat serta akurat dan informasi yang tersaji dapat tepat waktu pada saat di butuhkan.

Pengertian jasa menurut Lupiyoadi (2013:7), “Jasa adalah setiap tindakan ekonomi yang hasilnya tidak merupakan produk dalam bentuk fisik atau konstruksi, yang umumnya di konsumsi pada saat yang sama dengan waktu yang di hasilkan dan

memberikan nilai tambah (misalnya kenyamanan, hiburan, kesenangan, atau kesehatan) atau pemecahan masalah yang dihadapi oleh konsumen”.

Berdasarkan definisi tersebut dapat disimpulkan bahwa jasa adalah suatu tindakan atau suatu kegiatan yang mencakup semua aktifitas yang hasilnya berupa kinerja yang dapat di terima oleh konsumen.

2.1.1. Pengertian Sistem

Menurut Ladjamudin (2013:3) mendefinisikan “Sistem sebagai seperangkat elemen-elemen yang berintegrasi dengan maksud yang sama untuk mencapai tujuan bersama”.

Menurut Sutabri (2016:2) mengemukakan bahwa “Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi saling tergantung satu sama lain dan terpadu”.

Dengan demikian dapat disimpulkan bahwa sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berhubungan atau saling berinteraksi mempunyai ketergantungan satu dengan yang lainnya dan terpadu untuk mencapai suatu tujuan tertentu.

Model umum atau unsur-unsur yang mewakili suatu sistem terdiri dari masukan (*input*), pengolahan (*processing*) dan keluaran (*output*). Suatu sistem tidak lepas dari umpan balik (*feedback*) dilingkungan sekitarnya atau dari keluaran (*output*) tersebut. Setiap sistem terdiri dari struktur dan proses, dimana struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut, sedangkan proses sistem menjelaskan cara

kerja setiap unsur sistem tersebut dalam mencapai tujuan sistem. Sub sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri dari berbagai sistem yang lain atau lebih kecil.

Suatu sistem mempunyai tujuan (*goal*) dan sasaran (*objective*). Sasaran dari sistem sangat menentukan bentuk masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan. Sasaran merupakan hal-hal yang menjadi objek dan titik fokus untuk mencapai tujuan. Suatu sistem bisa dikatakan berhasil menjalankan fungsinya apabila berhasil mencapai sasaran (*objective*) dan tujuan (*goal*) dari sistem tersebut.

2.1.2. Karakteristik Sistem

Suatu sistem memiliki karakteristik atau sifat tertentu, karakteristik dari suatu sistem merupakan unsur-unsur yang terpadu untuk mencapai tujuan tertentu, Menurut Ladjamudin (2013:3-5), sistem mempunyai karakteristik sebagai berikut:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses

2. Batasan Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu

sistem dipandang sebagai satu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan di kendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media yang menghubungkan antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal. *Maintenance* input adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal *Input* adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

6. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.1.3. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lainnya, oleh karena itu sistem dapat diklasifikasikan beberapa sudut pandang. Klasifikasi sistem menurut Ladjamudin (2013:6) yaitu:

1. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical System*)

- a. Sistem Abstrak (*Abstract System*) adalah sistem yang berupa pemikiran atau ide yang tak tampak secara fisik.

Contoh:

Sistem teologi adalah sebuah susunan gagasan mengenai Tuhan, manusia dan alam.

b. Sistem Fisik (*Physical System*) adalah sistem yang ada secara fisik.

Contoh:

Sistem peredaran darah, sistem sekolah, sistem transportasi, dan sistem komputer.

2. Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Human Made System*).

a. Sistem Alamiah (*Natural System*) adalah sistem yang terjadi karena adanya proses alam.

Contoh:

Pergantian siang dan malam, erosi dan bencana alam.

b. Sistem Buatan Manusia (*Human Made System*) adalah sistem yang terjadi karena sengaja dibuat oleh manusia.

Contoh:

Sistem komputer dan sistem irigasi.

3. Sistem Tertentu (*Deterministik System*) dan Sistem Tidak Tentu (*Probabilistik System*).

a. Sistem Tertentu (*Deterministik System*) adalah suatu sistem yang beroperasi dengan tingkah yang dapat diprediksi.

Contoh:

Hasil pertandingan sepak bola dan prestasi

b. Sistem Tak Tentu (*Probabilistik System*) adalah sistem yang kondisi masa depannya tidak dapat diprediksi.

Contoh:



Sistem kematian.

4. Sistem Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*).

- a. Sistem Tertutup (*Closed System*) adalah sistem yang tidak berhubungan atau tidak dipengaruhi oleh lingkungan luarnya sistem ini bekerja otomatis tanpa campur tangan pihak luar.

Contoh

Sistem adat masyarakat baduy.

- b. Sistem Terbuka (*Open System*) adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya sistem ini menerima masukan dan menghasilkan pengeluaran untuk subsistem lainnya.

Contoh:

Sistem musyawarah.

2.1.4. Elemen Sistem

Menurut Kadir (2014:62-63) elemen yang membentuk sebuah sistem yaitu:

1. Tujuan

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terarah dan tidak terkendali.

2. Masukan

Masukan (*input*), sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk di proses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak.

3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

4. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

5. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feed-back*), yang mencuplik keluaran. Umpan balik ini di gunakan untuk mengendalikan baik masukan maupun proses. Tujuannya untuk mengukur agar sistem berjalan sesuai tujuan.

2.1.5. Daur Hidup Sistem (*System Development Life Cycle (SDLC)*)

Daur hidup sistem menurut Kadir (2014:344), “*System Development Life Cycle (SDLC)* merupakan metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi”. Pengertian lain menurut Rosa dan M. Shalahuddin (2016:26), “*System Development Life Cycle (SDLC)* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan cara-cara yang sudah teruji baik atau *best practice*”. Daur hidup sistem terdiri menjadi beberapa tahapan (*fase*) yaitu sebagai berikut:

1. Mengenali Adanya Kebutuhan

Sebelum segala sesuatunya terjadi, pasti terlebih dahulu adanya kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan suatu organisasi. Permintaan kebutuhan itu meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas, tanpa adanya kejelasan mengenai kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

2. Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisa kebutuhan yang dibutuhkan dengan membangun suatu sistem untuk memenuhi kebutuhan tersebut.

3. Penerapan Sistem

Setelah tahap pembangunan sistem selesai, kemudian sistem akan dioperasikan. Penerapan sistem merupakan tahap yang penting dalam daur ulang sistem, peralihan dari tahap pembangunan menuju tahap operasional adalah penerapan sistem, yang merupakan tahap akhir dari suatu pembangunan sistem.

4. Pengoperasian Sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi yang ditunjang oleh sistem informasi selalu mengalami perubahan karena pertumbuhan kegiatan, perubahan peraturan dan kebijaksanaan, ataupun kemajuan teknologi. Untuk mengatasi perubahan-prubahan tersebut, sistem harus diperbaiki atau diperbaharui.

5. Sistem Menjadi Tidak Layak

Perubahan-perubahan dengan kebutuhan yang begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan pada sistem yang sedang berjalan. Secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

2.1.6. Pengertian Informasi

Pengertian Informasi menurut Sutabri (2016:4), “Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan”. Sumber dari informasi adalah data. Data adalah kumpulan huruf atau angka yang belum diolah sehingga tidak memiliki arti. Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi yang tidak mempunyai makna atau tidak berpengaruh langsung kepada pemakai.

Sistem pengolah informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tidak berguna menjadi berguna bagi penerimanya. Nilai atau kualitas informasi berhubungan dengan keputusan. Apabila tidak ada pilihan atau keputusan, maka informasi menjadi tidak diperlukan. Karakteristik dari nilai atau kualitas suatu informasi sebagai berikut:

1. Akurat (*Accurate*)

Informasi yang diberikan harus mencerminkan keadaan yang sebenarnya, karena keakurasian sebuah informasi dapat menjadi tolak ukur ketepatan dan keberhasilan dalam pengambilan keputusan.

2. Tepat Waktu (*Timeliness*)

Informasi harus tersedia pada saat diperlukan dan tidak boleh terlambat karena didalam proses pengambilan keputusan, informasi yang sudah lama atau datanginya setelah suatu keputusan diambil tidak akan lagi memiliki nilai. Jadi, semakin baru (*up to date*) informasi tersebut maka akan semakin berguna. Ketepatan waktu sangat penting bagi datanginya informasi yang dibutuhkan oleh keadaan tertentu.

3. Lengkap

Informasi bagi para pengguna atau pembuat keputusan harus memperoleh informasi yang menyajikan suatu gambaran lengkap atas suatu masalah tertentu atau solusi yang dapat dijadikan landasan keputusan.

4. Relevan

Berhubungan dengan keputusan yang akan diambil, hal ini berhubungan dengan kelengkapan suatu informasi. Tingkat relevansi dikatakan sangat tipis karena tidak lengkapnya suatu informasi, maka informasi tersebut tidak ada gunanya. Relevansi suatu informasi akan menjadi penting karena hal itu bisa menjadi variabel-variabel yang menentukan pengambilan keputusan oleh organisasi.

5. Terpercaya

Isi informasi dapat dipercaya, hal ini tergantung kepada pemberi informasi.

6. Terverifikasi

Informasi dapat dilacak ke sumber lainnya, apabila pemakai laporan tidak yakin dengan informasi yang diterimanya, untuk mendapatkan sumber informasi yang diperoleh dan memeriksa apakah benar informasi yang telah diterimanya.

7. Mudah Dipahami

Informasi harus siap dipahami oleh penerimanya atau pengguna atau pembacanya, karena penerima informasi tidak ingin berpikir lagi dalam menerima informasi, untuk mengetahui kesimpulan dari informasi tersebut.

8. Mudah Diperoleh

Informasi yang sulit diperoleh bisa tidak berguna, apabila pengguna mendapatkan informasi yang sulit diperoleh.

2.1.7. Pengertian Sistem Informasi

Menurut Sutabri (2016:38), “Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu”.

Sistem Informasi merupakan suatu sistem yang ada didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi yang bersifat manajerial dan kegiatan strategis dari suatu organisasi dan pihak luar tertentu dengan laporan yang diperlukan.

2.2. Konsep Dasar Program

Konsep dasar program merupakan bagian penentuan kebutuhan dari sistem informasi yang dibutuhkan. Pada umumnya program merupakan sederetan instruksi atau statement yang dapat dimengerti oleh komputer. Instruksi tersebut berfungsi untuk

mengatur pekerjaan apa saja yang akan dilakukan oleh komputer agar mendapatkan dan menghasilkan suatu hasil atau keluaran yang diharapkan. Menurut Kadir (2014:192) memberi pengertian bahwa “Program adalah sekumpulan instruksi yang digunakan untuk mengatur perangkat keras komputer agar melaksanakan tindakan tertentu”. Sedangkan menurut Fadallah dkk (2018:61) mengungkapkan bahwa “Program adalah kumpulan instruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu”.

2.2.1. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek atau *Object Oriented Programming (OOP)* merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

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. Aplikasi saat ini sangat beragam karena tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi kesemua jenis aplikasi tersebut. Menurut Rossa dan M. Shalahuddin (2016:100) memberikan pengertian bahwa “Metodelogi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya”.

Sedangkan menurut Kadir (2014:204) mengemukakan bahwa “Pemrograman berorientasi objek adalah mengombinasikan data dan prosedur-prosedur untuk mengakses data menjadi sebuah kesatuan unit”.

Objek merupakan entitas yang memiliki atribut, karakter (*behaviour*) dengan disertai kondisi (*state*). Objek merepresentasikan sesuatu sistem yang nyata, seperti siswa, sensor atau mesin atau sesuatu dalam bentuk konsep seperti nasabah bank. Kelas (*Class*) merupakan penggambaran satu-kesatuan objek yang memiliki atribut dan karakter (*behaviour*) yang sama, Kelas dan objek merupakan jantung dari pemrograman berorientasi objek

2.2.2. Bahasa Pemrograman Visual Studio 2010

Bahasa pemrograman komputer merupakan sarana komunikasi yang menghubungkan antara manusia dengan komputer. Bahasa pemrograman komputer dikelompokkan menjadi dua kelompok besar, yaitu bahasa pemrograman tingkat rendah (*Low level language*) dan bahasa pemrograman tingkat tinggi (*High level language*). Dengan bahasa pemrograman komputer, manusia dapat membuat sebuah program untuk menyelesaikan suatu masalah.

Menurut Kadir (2014:192) memberikan pengertian bahwa “Bahasa pemrograman adalah sekumpulan instruksi yang ditunjukkan agar orang bisa menuangkan perintah yang nantinya akan dijalankan oleh komputer”.

Menurut Yesputra (2017:1), “Visual Studio 2010 merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang

memiliki berbagai macam tipe antara lain aplikasi desktop (*Windows Form*, *commandLine*, (*Console*), aplikasi web, *windows mobile* (poket PC)”.

Visual Studio terbagi menjadi beberapa tipe menurut Yesputra (2017:1) diantaranya:

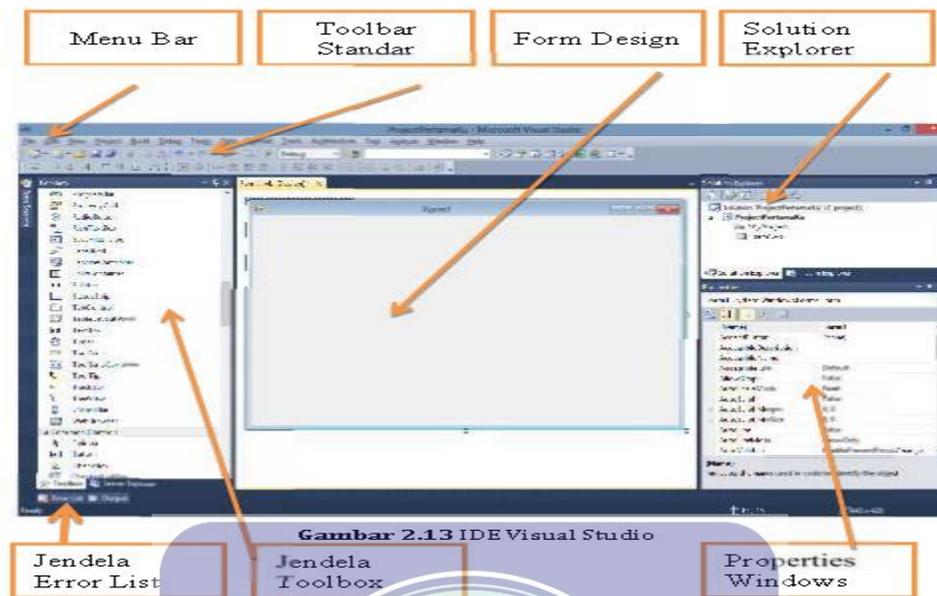
1. Visual Studio 2010 Express Edition
2. Visual Studio Standard Edition
3. Visual Studio 2010 Professional Edition
4. Visual Studio Ultimate Edition

1. Komponen Visual Basic .Net

Komponen yang digunakan pada *visual basic* Dalam pemograman berbasis *Object Oriented Programming (OOP)*, sebuah program dibagi menjadi bagian-bagian kecil yang disebut dengan obyek. Setiap obyek memiliki *entity* terpisah dengan *entity-entity* lain dalam lingkungannya. Obyek-obyek yang terpisah ini dapat diolah sendiri-sendiri, dan setiap obyek memiliki sekumpulan sifat dan metode yang melakukan fungsi tertentu sesuai dengan yang dibuat.

a. *Integrated Development Environment (IDE) Visual Studio*

IDE merupakan lingkungan tempat kita mengembangkan sistem yang sudah terintegrasi antara Kompiler, Desain, Koding dan lainnya. Dengan menggunakan IDE ini pekerjaan kita sebagai *programmer* akan mudah dan juga manajemen proyek yang sedang kita kembangkan sangat efisien di lakukan.



Gambar 2.13 IDE Visual Studio

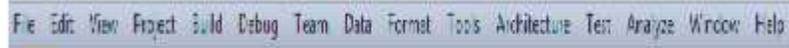
Sumber: Yesputra (2017:18)

Gambar II.1 IDE Visual Studio

Penjelasan dari beberapa item yang sudah di tandai di atas, yang terdapat di dalam jendela *visual basic*, elemen tersebut merupakan elemen yang sering di gunakan dalam memprogram aplikasi. Untuk lebih mudah memahaminya akan di paparkan dalam pembahasan berikut ini.

1) Menu Bar

Menu bar merupakan suatu baris *menu* yang mempunyai 11 menu utama, masing-masing memilik *sub menu* dan perintah lengkap dengan *shortcut key*



Sumber: Yesputra (2017:19)

Gambar II.2
Menu Bar Visual Studio

2) *Toolbar standar*

merupakan suatu baris *menu* yang mempunyai fungsi yang sama pada setiap *tool standard* pada umumnya. Seperti fungsi untuk menyimpan dan menambah *project* baru, mengatur tampilan program dan masih banyak yang lainnya.

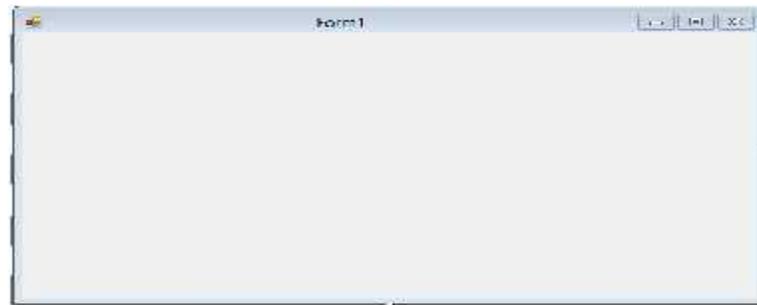


Sumber: Yesputra (2017:19)

Gambar II.3
Tollbar Standar Visual Studio

3) *Form Design*

Form design merupakan suatu lembar form yang berfungsi untuk merancang tampilan aplikasi secara visual dengan menempatkan kontrol-kontrol yang diperlukan.



Sumber: Yesputra (2017:19)

Gambar:II.4
Form Design Visual Studio

4) *Toolbox*

Merupakan suatu jendela yang berfungsi untuk menampung komponen-komponen standar.

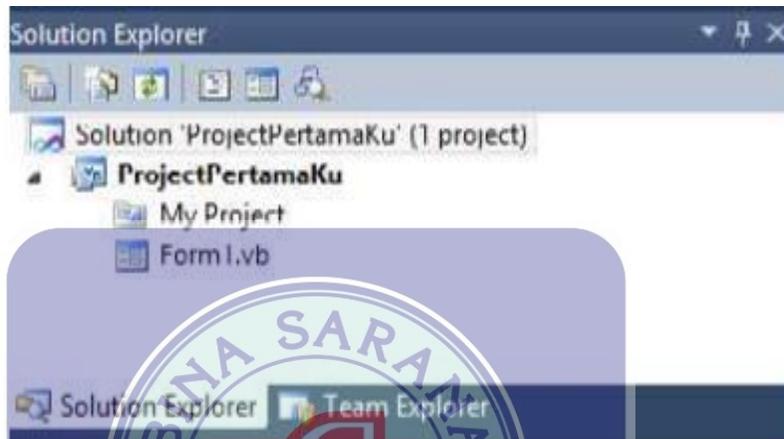


Sumber: Yesputra (2017:20)

Gambar: II.5
Toolbox Visual Studio

5) *Solution Explorer*

Solution Explorer merupakan jendela yang berfungsi untuk menampilkan objek yang akan anda gunakan untuk membuat aplikasi seperti form, class, dan objek lainnya.

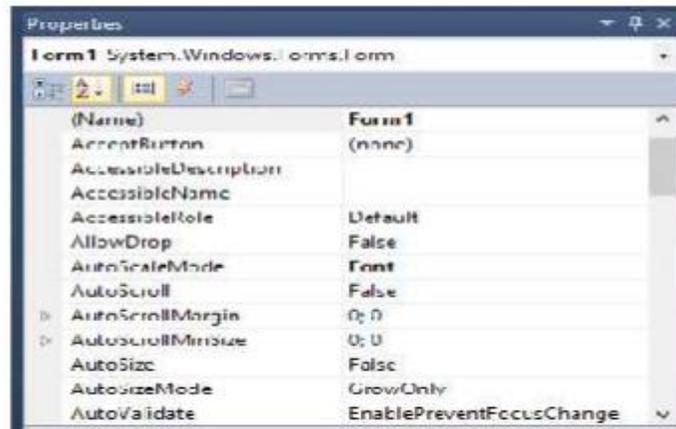


Sumber: Yesputra (2017:21)

Gambar: II.6
Solution Explorer Visual Studio

6) *Proferties Windows*

Proferties Windows merupakan suatu jendela yang berfungsi untuk mengatur nilai *proferties* dari masing-masing komponen yang akan di gunakan.



Sumber: Yesputra (2017:21)

Gambar: II.7
Properties Windows Visual Studio

7) Error List

Error list merupakan suatu jendela yang di gunakan untuk menampilkan setiap kesalahan dari pembuatan kode program suatu aplikasi.



Sumber: Yesputra (2017:22)

Gambar: II.8
Error List Visual Studio

2.2.3. Basis Data

Menurut Kadir (2014:218), “Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data di maksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas”.

Untuk mengelola basis data di perlukan perangkat lunak yang di sebut *Database Management System* (DBMS). DBMS menurut Kadir (2014:218) ”Adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, mengakses basis data dengan cara yang peraktis dan efisien”.

2.2.4. Aplikasi Basis Data (*Database*) *My Structure Query Language* (*MySQL*)

Aplikasi basis data sering digunakan oleh para pembuat aplikasi sebagai media pengolahan basis data. Aplikasi basis data yang sering digunakan dalam pengolahan basis data yaitu *My Structure Query Language* (*MySQL*). Salah satu aplikasi basis data yang sering digunakan untuk mengolah dan menata file-file yaitu *MySQL*.

Menurut Sadeli (2013:10) memberikan pengertian bahwa “*MySQL* adalah *database* yang menghubungkan *script php* menggunakan perintah *query* dan *escaps character* yang sama dengan *php*”. Menurut Madcoms (2013), “*Mysql* merupakan *databases* yang sangat populer. Beberapa keuntungan yang dimiliki *mysql* yaitu: bersifat *open source*, menggunakan bahasa *Structure Query Language* (*SQL*), *super perfomance* dan *reliable*, mudah dipelajari, mampu bekerja dilintas *platform* dan *multi user*.”

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya. *MySQL* dikembangkan oleh perusahaan swedia bernama *MySQ LAB* yang pada saat ini bernama *Tex Data Konsult AB* sekitar tahun 1994-1995.

MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk *internet* (*PHP* dan *Perl*). *MySQL* dan *PHP* dianggap sebagai pasangan *software* pembangun aplikasi *web* yang ideal. *MySQL* lebih sering digunakan untuk membangun aplikasi berbasis *web*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script PHP*. Adapun kelebihan-kelebihan dari *MySQL* yaitu:

1. *Source MySQL* dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan basis data dapat dilakukan dengan mudah.
4. *MySQL* merupakan program yang *multithreaded*, sehingga dipasang pada *server* yang memiliki multi CPU.
5. Didukung program-program umum seperti *C*, *C++*, *Java*, *Perl*, *PHP*, *Python* dan sebagainya.
6. Bekerja pada berbagai *platform* yaitu tersedia berbagai versi untuk berbagai sistem operasi.
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
8. Mendukung *ODBC* untuk sistem operasi *Windows*.

Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

2.2.5. Metode Pengembangan Perangkat Lunak Model *Waterfall*

Menurut Rosa (2016:28-30), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut di mulai analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)”.

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*, Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu di dokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya, Desain perangkat lunak yang di hasilkan pada tahap ini juga perlu di dokumentasikan.

3. Pembuatan kode program

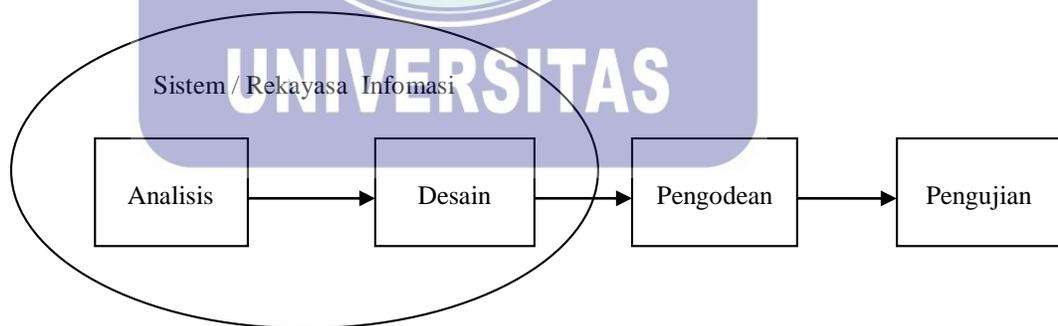
Desain harus di translasikan pada program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah di buat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional dan memastikan bahwa semua bagian telah di uji. Hal ini di lakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang di hasilkan sesuai dengan yang di inginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah di kirimkan ke *user*, Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.



Gambar II.9
Ilustrasi Model Waterfall

2.3. Teori Pendukung

2.3.1. *Entity Relationship Diagram (ERD)*

Menurut Ladjamudin (2013:142), “*Entity Relationship Diagram (ERD)* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

Entity Relationship Diagram (ERD) merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. Dengan menggunakan *Entity Relationship Diagram* dapat memodelkan struktur data dan hubungan antar data yang begitu kompleks, sehingga mudah untuk dibaca dan dimengerti, dalam pemodelan struktur data dapat mengabaikan proses yang harus dilakukan, selain itu pemodelan struktur data dengan menggunakan *Entity Relationship Diagram (ERD)* dapat menjawab pertanyaan seperti, data apa saja yang kita butuhkan dan bagaimana data yang satu berhubungan dengan data yang lainnya

Derajat kardinalitas yang terdapat dalam *Entity Relationship Diagram* adalah:

1. Satu ke Satu (*One to One*) (1:1)

Adalah derajat kardinalitas yang menunjukkan adanya relasi himpunan entitas yang satu dengan satu entitas lainnya.

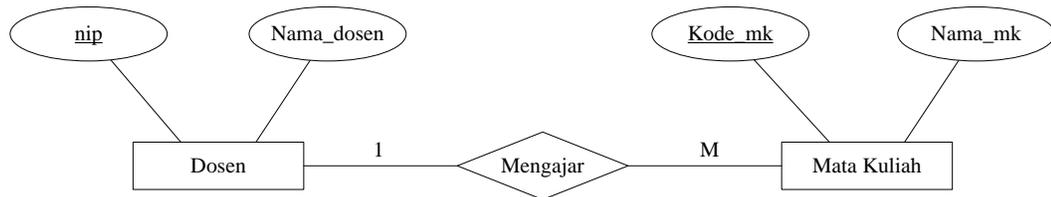


Sumber: Rosa dan Shalahuddin (2016:166)

Gambar II.10.
Contoh Derajat Kardinalitas Satu ke Satu (*One to One*) (1:1)

2. Satu ke Banyak (*One to Many*) (1:N)

Adalah derajat kardinalitas yang menunjukkan adanya relasi antar himpunan entitas yang satu dengan entitas banyak entitas lainnya.

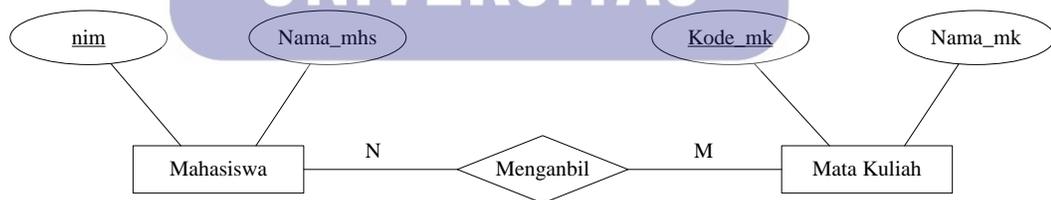


Sumber: Rosa dan Shalahuddin (2016:166)

Gambar II.11.
Contoh Derajat Kardinalitas Satu ke Banyak (*One to Many*) (1:N)

3. Banyak ke Banyak (*Many to Many*) (N:M)

Adalah derajat kardinalitas yang menunjukkan adanya relasi antar himpunan banyak entitas dengan banyak entitas lainnya. Jika terjadi relasi many to many maka akan menghasilkan sebuah entitas baru.



Sumber: Rosa dan Shalahuddin (2016:166)

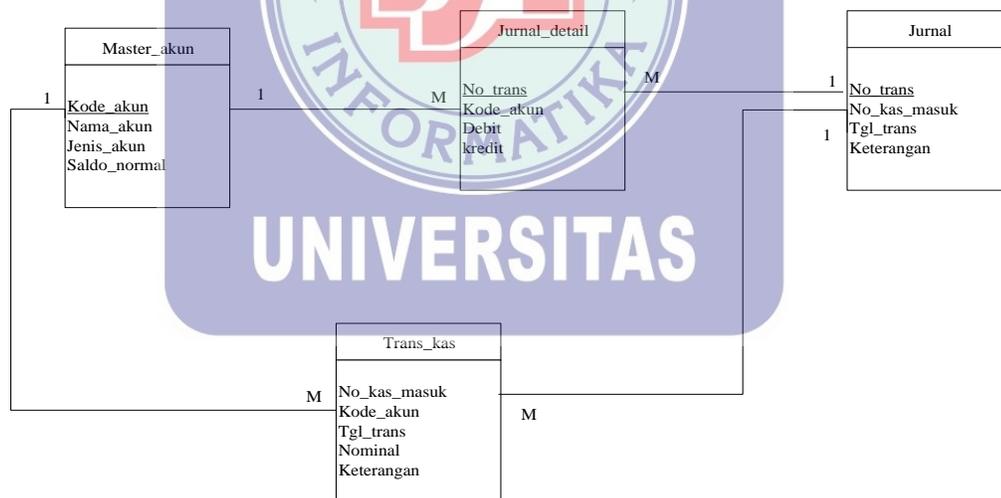
Gambar II.12.
Contoh Derajat Kardinalitas Banyak ke Banyak (*Many to Many*) (N:M)

2.3.2. Logical Record Structure (LRS)

Logical Record Structure (LRS) adalah representasi dari *structure record-record* pada tabel-tabel yang berbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah tabel dan *Foreign Key (FK)*. Aturan pembuatan *Logical Record Structure (LRS)* yaitu:

1. Setiap entitas akan diubah menjadi bentuk kotak.
2. Sebuah atribut disatukan dalam sebuah kotak bersama entitas.
3. Sebuah relasi dipisah dalam sebuah kotak tersendiri atau menjadi entitas baru dalam bentuk belah ketupat.

Berikut contoh dari *Logical Record Structure*:



Sumber: Syara, Chintya (2018).

Gambar II.13.
Logical Record Structure

2.3.3. *Unified Modelling Language*

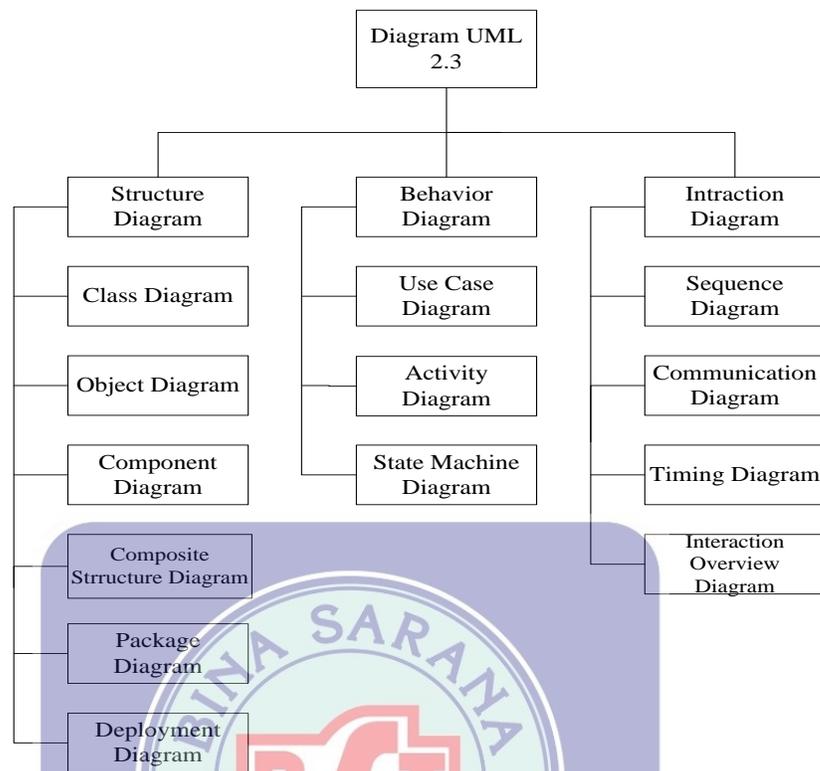
Pemodelan menggunakan *Unified Modeling Language (UML)* merupakan metode pemodelan berorientasi objek dan berbasis *visual*. Karenanya pemodelan menggunakan *UML* merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis dari pada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional.

Menurut Rosa dan Shalahuddin (2016:133), “*Unified Modeling Language (UML)* salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requierment*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman objek”.

Secara fisik, UML merupakan sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML, terbaru adalah 2.3 yang terdiri dari empat macam spesifikasi, yaitu *Diagram Interchange Specification*, *UML Infrastructure*, *UML Superstructure*, dan *Object Constraint Language (OCL)*.

1. Diagram *Unified Modeling Language (UML)*

Pada UML 2.3 terdiri dari tiga belas (13) macam diagram yang dikelompokkan kedalam tiga kategori. Adapun pembagian kategori dan macam-macam diagram tersebut akan digambarkan pada diagram berikut.



Sumber: Rosa dan Shalahuddin (2016:140)

Gambar II.14
Diagram Unified Modeling Language (UML)

Secara garis besar penjelasan gambar diatas adalah sebagai berikut:

- a. *Structure Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. *Behavior Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- c. *Interaction Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antara sub sistem pada suatu sistem.

2. Use Case Diagram

Menurut Rosa dan Shalahuddin (2016:155), “*Use Case Diagram* merupakan pemodelan untuk kelakuann (*behavior*) sistem informasi yang aan dibuat. *Use Case* mendeskripsi sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut adalah simbol-simbol yang terdapat didalam *use case diagram*.



Sumber: Dermawan dan Hartini (2017)

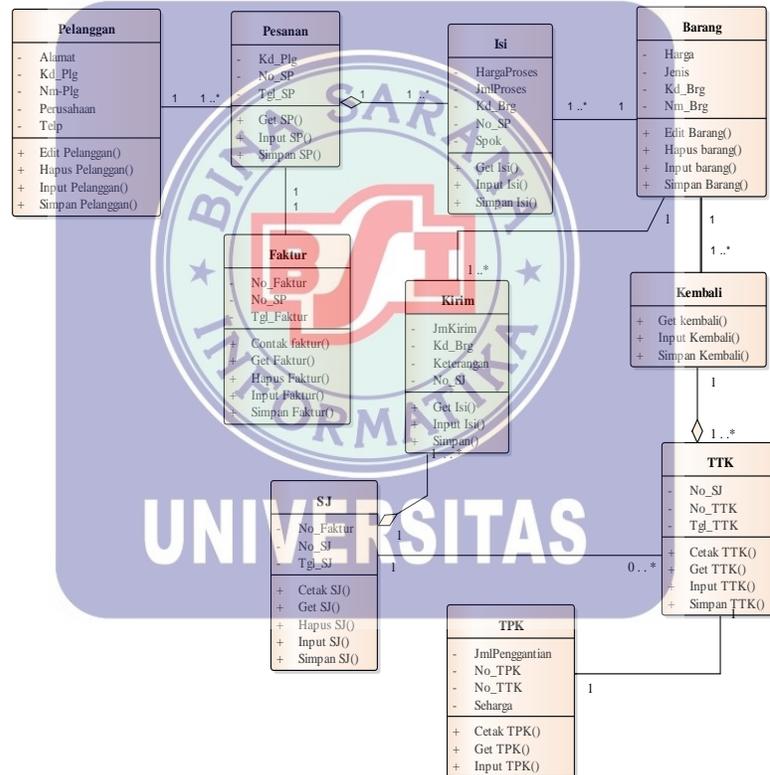
Gambar II.15.
Contoh Use Case Diagram

3. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, dimana kelas ini memiliki apa yang disebut dengan atribut atau variabel-variabel yang dimiliki oleh suatu kelas dan operasi atau metode yang berarti fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuat atau *programmer* membuat kelas-kelas yang sesuai dengan rancangan didalam diagram kelas sehingga terjadi kesesuaian antara dokumentasi perancangan dan perangkat lunak, dengan demikian kelas-kelas yang ada pada struktur sistem dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem tersebut.

Berikut adalah simbol-simbol yang terdapat dalam kelas diagram:

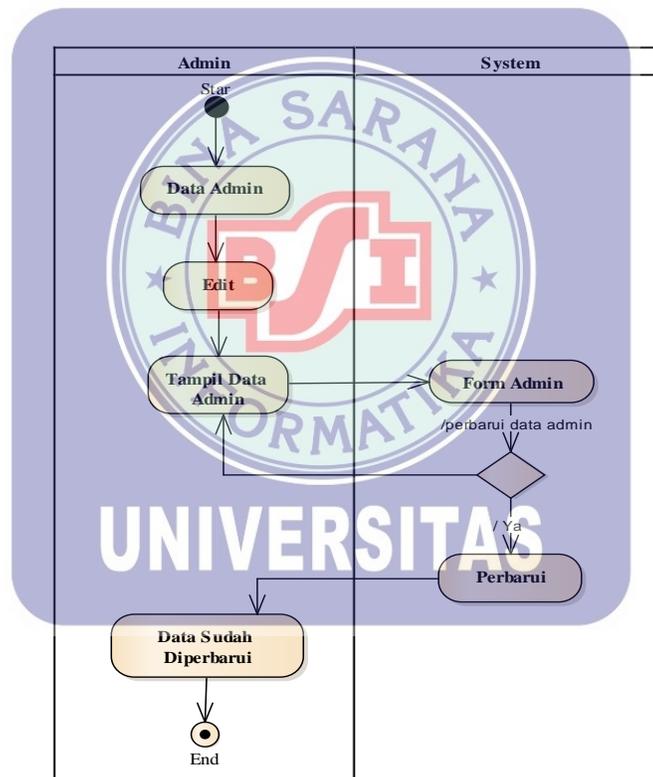


Sumber: Sunarti (2014)

Gambar II.16.
Class Diagram

4. Activity Diagram

Menurut Rosa dan Shalahuddin (2016:157), “*Activity diagram* atau diagram aktivitas adalah *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

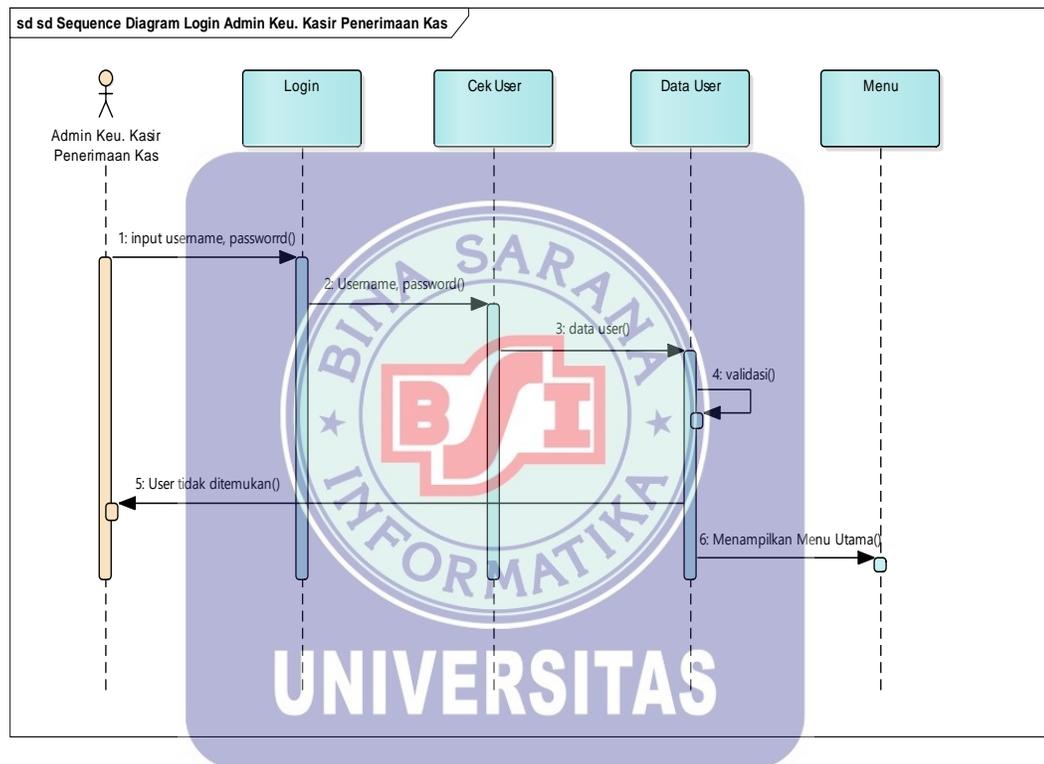


Sumber: Dermawan dan Hartini (2017)

Gambar II.17.
Contoh Activity Diagram

5. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu, dalam menggambar diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* terlebih dahulu.



Sumber: Syara, Chintya (2018).

Gambar II.18.
Contoh Sequence Diagram