BAB II

TINJAUAN PUSTAKA

Pada bagian ini akan dibahas mengenai teori-teori yang berkaitan dengan penelitian ini. Teori-teori ini dibagi menjadi dua jenis, yaitu teori umum, dan teori khusus.

2.1. Tinjauan Teori Umum

Dalam bagian ini, penulis akan menjelaskan mengenai teori–teori umum menurut pendapat ahli.

2.1.1. Pengertian Sistem Informasi

(Husda, 2012) mengatakan, "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

Sistem berasal dari bahasa Latin dan bahasa Yunani adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan

untuk menggambarkan suatu set entitas yang beinteraksi, di mana suatu model matematka seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki *item-item* penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu Negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada di negara tersebut. Contoh lainnya adalah sistem komputer yang terdiri dari: *software*, *hardware*, *brainware*, sistem akuntansi, dan lain-lain.

Syarat-syarat yang harus dipenuhi oleh sistem menurut (Husda, 2012), yaitu:

1. Sistem harus dibentuk untuk menyelesaikan tujuan.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi, dan material) lebih penting daripada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

Karakteristik yang dimiliki oleh sistem adalah:

1. Komponen (*Component*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-

komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut dengan subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

2. Batas Sistem (*Boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan yang lainnya berbeda tetapi tetap saling beinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut (Husda, 2012).

3. Lingkungan Luar Sistem (*Environment*)

   Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar

yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

   Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain *output* dari suatu subsistem akan menjadi *input* dari subsistem lainnya.

5. Masukkan Sistem (*Input*)

   Merupakan energi yang dimasukkan ke dalam sistem. Masukkan dapat berupa:

   a. Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi.
   b. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

   Sebagai contoh didalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi (Husda, 2012).

6. Keluaran Sistem (*Output*)

   Merupakan hasil dari energi yang diolah dari sistem. Meliputi:

   a. Keluaran yang berguna, contohnya informasi yang dikeluarkan oleh komputer.

b. Keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

7. Pengolah Sistem (*Process*)

   Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh *CPU* pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi, bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Tujuan Sistem (*Goal*)

   Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

(Husda, 2012) juga mengatakan bahwa, sistem juga mempunyai beberapa klasifikasi, yaitu:

1. Sistem Abstrak (*Abstract System*)

   Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (Sistem *Teologia* yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan Manusia).

2. Sistem Fisik (*Physical System*)

   Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya (sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain).

3. Sistem Alamiah (*Natural System*)

   Sistem yang terjadi melalui proses alam dalam artian tidak dibuat oleh manusia (sistem tata surya, sistem galaksi, sistem reproduksi, dan lain-lain).

4. Sistem Buatan Manusia (*Human Made System*)

   Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system* (contoh sistem informasi).

5. Sistem Tertentu (*Deterministic System*)

   Sistem beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh: sistem komputer).

6. Sistem Tak Tentu (*Probabilistic System*)

   Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas (contoh: sistem manusia).

7. Sistem Tertutup (*Close System*)

   Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup) (Husda, 2012).

8. Sistem Terbuka (*Open System*)

   Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomisasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern (Husda, 2012).

Secara umum sistem juga dapat didefinisikan sebagai kumpulan objek, ide, berikut keterkaitannya di dalam mencapai tujuan. Dengan kata lain, sistem adalah kumpulan komponen (sub-sistem fisik dan non-fisik / logika) yang saling berhubungan satu sama lainnya dan bekerja sama untuk mencapai suatu tujuan. Istilah "sub-sistem" cenderung digunakan untuk memudahkan analisis suatu sistem beserta pengkomunikasiannya. Walaupun demikian, jika kompleksitas sistemnya dipandang cukup tinggi, maka sistemnya dibagi lagi ke dalam beberapa sub-sistem untuk memenuhi tujuan kesederhanaan, kemudahan pemahaman, pengendalian, dan implementasinya. Dengan demikian, suatu (model) sistem yang memiliki tingkat kompleksitas yang tinggi pun dapat dipahami lebih mudah (Prahasta, 2014).

(Husda, 2012) menyimpulkan, di dalam suatu organisasi atau perusahaan, informasi merupakan sesuatu yang memiliki arti yang sangat penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Secara umum informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya.

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus (Husda, 2012).

Adapun fungsi-fungsi informasi adalah sebagai berikut:

a. Untuk meningkatkan pengetahuan bagi si pemakai.
b. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
c. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

Informasi yang berkualitas harus akurat, tepat pada waktunya dan relevan.

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat waktu

   Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah using tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal bagi organisasi. Saat ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu didapat sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkannya.

3. Relevan

   Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda-beda (Husda, 2012).

(Prahasta, 2014) menjelaskan, istilah “data” dan “informasi” sering digunakan bergantian dan tertukar meskipun kedua istilah ini sebenarnya merujuk pada konsep yang berbeda. Data merupakan bahasa, *mathematical*, dan atau simbol pengganti lain yang disepakati secara umum dalam menggambarkan suatu simbol pengganti lain yang disepakati secara umum dalam menggambarkan suatu

objek, manusia, peristiwa, aktivitas, konsep, atau objek penting lainnya. Singkatnya, data merupakan suatu kenyataan apa adanya. Sedangkan informasi adalah data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya.

Contoh umum yang mudah dipahami mengenai penggunaan terminology data dan informasi banyak ditemukan dalam konteks perusahaan. Pada konteks ini, kunci transaksi yang paling sederhana adalah kertas bon pesanan dari pelanggan. Nilai informasi pesanan ini masih bergantung pada tingkatan karyawan yang menanganinya. Mereka yang menangani bon pesanan (petugas kredit, pergudangan, pengiriman) melihat isi bon sebagai informasi yang diperlukan untuk melakukan pekerjaan masing-masing.

Seorang *sales* tertarik dengan bon-bon yang berhubungan dengan pelanggannya beserta waktu pesanan tersebut diberikan dan kapan harus dipenuhi. Manajer *sales* juga tertarik dan memberikan nilai yang tinggi pada data bon pesanan dari para pelanggannya, tetapi ketika dipresentasikan sebagai referensi suatu kuota, prediksi, atau anggaran. Seorang akuntan memandang pesanan ini sebagai data hingga diproses dan disajikan dalam bentuk laporan bulanan (Prahasta, 2014).

Bon pesanan dari pelanggan seperti di atas merupakan contoh data dalam bentuknya yang sederhana. Sementara itu, hal mengenai apakah data ini akan dinilai sebagai informasi atau tidak, akan bergantung pada situasi dan kondisi setiap penerimanya. Situasi dan kondisi yang dimaksud bisa mencakup kebiasaan, tingkah laku, emosi, tujuan hidup, pekerjaan,dan tugas individu. Jumlah data yang

tersedia setiap saat bisa saja tidak terbatas. Kecuali jika yang bersangkutan telah mengidentifikasi dengan baik apa saja kebutuhan informasi yang harus dipenuhi dan dapat ditanganinya. Oleh karena itu, jika penyediaan informasinya lebih dipentingkan daripada data itu sendiri, maka diperlukan analisis situasi berikut penentuan kebutuhan informasinya yang lebih spesifik.

Sistem informasi didasarkan pada asumsi bahwa perancangnya dapat mengidentifikasi kebutuhan informasinya. Selain itu, ia juga dapat menentukan metode yang digunakan untuk menghasilkan informasi dari data. Dalam kaitan ini, kemudian muncullah terminology "informasi formal" dan "informasi non-formal". Contoh informasi formal mencakup peraturan pemerintah (PP), undang-undang (UU), surat keputusan (SK), surat perintah (SP), surat perjanjian atau kontrak, prosedur akuntansi, persyaratan perencanaan, anggaran dasar organisasi, permintaan pekerjaan, kebutuhan pengendalian dan kontrol, dan proses pengambilan keputusan umum. Sedangkan surat bukti pembayaran, surat tagihan (*invoices*), faktur, bon pembelian, tiket, dan sejenisnya merupakan contoh nyata informasi formal dalam bentuk yang telah terstruktur. Sementara itu, contoh informasi non-formal mencakup pendapat individu, intuisi firasat, prasangka, dugaan, kabar, isu, desas-desus, pengalaman pribadi, selentingan, *rules of thumb*, gosip, anggapan atau asumsi, dan lain sejenisnya (Prahasta, 2014).

Dari contoh di atas, nampak perbedaan antara informasi formal dan informasi non-formal. Informasi formal memungkinkan para penggunanya untuk mengekstrak, memproses, mentransformasikan, atau melakukan prosedur lainnya untuk menghasilkan informasi dari data. Di lain pihak, nilai informasi non-formal

juga bergantung pada interpretasi penggunanya, mau diterima atau tidak. Bentuk, wujud, dan kandungan informasi non-formal cenderung subjektif dan tidak terstruktur, dan proses konversi dari data hingga menjadi informasi tidak terpisahkan dari penerimanya. Kedua tipe informasi ini mungkin diperlukan dalam pengelolaan dan operasi suatu organisasi, tetapi informasi formal merupakan satu-satunya *output* yang valid dari sebuah sistem informasi.

(Prahasta, 2014) mengatakan, banyak atribut atau kualitas yang berkaitan dengan konsep informasi dapat membantu perancang dalam mengidentifikasi kebutuhan informasi, diantaranya adalah:

1. Akurat, tingkat kebebasan informasi dari kesalahan.
2. Presisi, ukuran detail untuk menyediakan informasi.
3. Tepat waktu, penerimaan informasi masih dalam toleransi waktu yang dibutuhkan oleh penerima, tidak kadaluarsa.
4. Jelas, tingkat kebebasan informasi dari keraguan.
5. Dibutuhkan, tingkat relevansi informasi dengan kebutuhan pengguna.
6. *Quantifiable*, tingkat kemampuan menyatakan informasi dalam bentuk numerik.
7. *Verifiable*, tingkat kesamaan nilai sebagai hasil pengujian informasi yang sama oleh berbagai pengguna.
8. *Accessible*, tingkat kemudahan dan kecepatan memperoleh informasi.
9. *Non-bias*, tingkat perubahan untuk memodifikasi informasi dengan tujuan mempengaruhi para penerimanya.
10. *Comprehensive*, tingkat kelengkapan informasi (Prahasta, 2014).

Pada saat beralih dari konsep penyediaan informasi yang tergeneralisasi penentuan nilai atribut informasi menjadi perlu. Hal ini tidak mudah. Sebagian atribut sulit dinyatakan secara objektif. Misalnya, seseorang bertanggung-jawab menjadwalkan keberangkatan dan kedatangan kapal. Salah satu tanggung-jawabnya adalah memantau posisi kapal. Ia akan ditanya mengenai posisi kapal secara rutin. Kemungkinan jawaban pertanyaannya adalah “masih di laut”, “dalam perjalanan menuju pelabuhan Belawan”, “satu hari menjelang Surabaya”, “bujur 100^{o} BT dan lintang 03^{o} LU”.

Meskipun responnya akurat dan tepat waktu, jawaban yang diberikan bisa tidak memenuhi kebutuhan si penanya karena salah satu atributnya gagal dipenuhi. Sifat respons mensyaratkan bahwa kita menentukan setiap “konteks yang penuh arti” dari individu dan kita juga menentukan relevansi setiap atribut ke dalam konteksnya. Jika penilaiannya benar, maka kita menyediakan informasi yang benar pula. Sebaliknya, jika kita salah dalam menentukan kebutuhan informasi (gagal mempertimbangkan kepentingan atribut), maka kita pun menyediakan data, tetapi akan disusul pertanyaan berikutnya yang antara lain adalah “bisa lebih jelas lagi?”, “dapatkah anda lebih rinci lagi?”, “dapatkah saya mengetahui posisi sebetulnya?”, “sedang di Jakarta atau di Jogjakarta tepatnya?”

Pada saat mengidentifikasi kebutuhan informasi, sedapat mungkin kebutuhan tersebut dideskripsikan dalam terminologi atribut informasi. Analisis yang benar akan menunjukkan keterkaitan erat antara kebutuhan informasi beserta atributnya pada perancangan sistem informasi. Jadi, yang diperlukan adalah penyediaan informasi yang benar pada waktu yang tepat (Prahasta, 2014).

Menurut (Prahasta, 2014), pada dasarnya, data perlu diproses sebelum menjadi informasi. Jika prosesnya kompleks, kompleksitasnya bisa direduksi dengan memecahkan menjadi beberapa sub-proses yang lebih sederhana tanpa memperhatikan mekanismenya, kita bisa mengidentifikasi sepuluh langkah transformasi data menjadi informasi. Setiap operasi atau kombinasinya bisa menghasilkan informasi dari data. Operasi-operasi tersebut adalah:

1. *Capturing*

   Operasi ini merupakan proses perekaman data dari fenomena alam, peristiwa, atau kejadian ke bentuk formulir seperti halnya formulir ukur, slip penjualan, daftar isian data pribadi, dan pesanan pelanggan. Di bidang geodasi, sebagai contoh, operasi ini terjadi saat surveyor mengukur dengan alat *theodolite* dan mencatat hasilnya (benang atas, benang bawah, benang tengah, dan tinggi alat) pada formulir kasar.

2. *Verifying*

   Operasi ini merupakan pemeriksaan data untuk memastikan bahwa datanya direkam dengan benar. Di bidang geodasi, operasi ini dilakukan dengan memeriksa formulir ukur, apakah ada kesalahan penulisan, salah ukur, atau kesalahan lain yang menyebabkan “salah-penutupnya” melebihi toleransi. Nilai di luar toleransi bisa mengakibatkan pengukuran diulang, seluruhnya / hanya sebagian saja, membengkaknya sumber daya waktu, tenaga, logistik, dan biaya.

3. *Classifying*

   Operasi ini menempatkan elemen data ke dalam klasifikasi yang lebih berarti pada penggunanya. Misalnya, data penjualan dapat diklasifikasikan menjadi tipe, ukran inventori, pelanggan, dan *salesperson*.

4. *Arranging*

   Operasi ini menempatkan elemen data sesuai dengan ukuran tertentu. Sebagai contoh, *file* / tabel inventori bisa diurutkan menurut *field* tingkat aktivitas, nilai, atau oleh atribut lain yang dikodekan di dalam tabelnya.

5. *Summarizing*

   Operasi ini mengumpulkan elemen data ke dalam suatu cara. Pertama, operasi ini mengakumulasikan data secara matematis. Kedua, operasi ini mereduksi data secara logis (Prahasta, 2014).

6. *Calculating*

   Operasi ini memerlukan pemanipulasian data secara aritmetik dan logika, menghasilkan informasi dari hasil hitungan data. Sebagai contoh, proses hitungan dilakukan untuk menghasilkan gaji pegawai (berdasarkan pangkat, golongan, gaji pokok, masa kerja, dan tanggungan).

7. *Storing*

   Operasi ini meletakkan data pada media penyimpanan lain (berbeda dengan media sumber datanya) seperti halnya kertas, *microfilm*, disket, *harddisk*, dan *CD*. Di bidang geodesi, operasi ini dilakukan ketika personil meng-*entry* data pada formulir ukur ke dalam file lembar-kerja

untuk disimpan secara permanen, diperiksa, dan dihitung kembali (Prahasta, 2014).

8. *Retrieving*

   Operasi ini memerlukan fasilitas akses ke elemen data yang sebelumnya tersimpan di media penyimpanannya. Di bidang geodesi, operasi ini dilakukan ketika surveyor memanggil kembali formulir ukurnya (*file*) untuk memeriksa kembali, meng-*edit*-nya, atau dibaca oleh program aplikasi lainnya.

9. *Reproducing*

   Operasi ini menduplikasikan (bisa mencakup *reproduce* / *print* / *copy*) data dari suatu media ke media lainnya. Operasi ini bisa diwujudkan dalam bentuk meng-*copy file* sehingga menghasilkan *file* lain, atau mencetak *file layout* peta hingga dihasilkan lembaran peta dalam bentuk analog.

10. *Communicating* / *Disseminating*

    Operasi ini mentransfer data dari suatu tempat ke tempat lainnya. Di bidang geodesi, operasi ini bisa diwujudkan dalam bentuk pendistribusian data (peta) kepada yang memerlukannya, baik secara *online* maupun secara fisik.

Aliran fakta simbol, *string* alfabetis / kumpulan angka rekaman kejadian disebut sebagai “data”. Pada diagram, aliran masuk ke proses. Oleh karena itu, data disebut masukan / *input*. Setelah aliran ini diproses (hingga menghasilkan keluaran dalam bentuk yang dimengerti), bentuk ini disebut “informasi”. Oleh

karena itu, dipandang dari sisi pemroses, data dikenal sebagai *input*, sementara informasi disebut sebagai *output*.

Di sisi pemrosesan, fenomena *input*-proses-*output* bersifat relatif, suatu proses bisa menerima *input* hasil olahan (*output*) proses lainnya. Hal ini nampak pada sekelompok manusia dengan perannya sebagai pemroses, pihak mana yang melakukan proses pertama, dan pihak lain mana yang melakukan proses berikutnya. Tidak jarang, sekelompok proses tidak hanya berkemampuan melakukan pemrosesan yang sekarang menjadi tugasnya, mereka juga mampu melakukan pemrosesan yang sebenarnya menjadi tugas kelompok lain pada saat ini. Selain itu, seiring dengan perubahan waktu dan proses belajar, suatu kelompok juga bisa melakukan proses lainnya. Oleh karena itu, kemampuan ganda dan daya-adaptasi manusia bisa "mengaburkan" pengertian data dan informasi. Untuk memperjelas hal ini diperlukan ketegasan peran pada rentang waktu tertentu (Prahasta, 2014).

Sebagai ilustrasi, sebagian praktisi geodesi bisa dibagi menjadi beberapa kelompok: survei, pengolahan citra dijital, kartografi, jaringan (*networking*), basis data, dan aplikasi.

a. Kelompok survei bertugas untuk:
    1. Melakukan survei dengan *theodolite*, *total-station*, *GPS* (*optional*), atau *waterpass*.
    2. Membuat sketsa lokasi dan progres kemajuan survey.
    3. Menghitung koordinat ukuran.
    4. Membuat peta manuskrip.

b. Kelompok pengolahan citra dijital bertugas untuk:
    1. Melakukan survei koordinat (ground control points) untuk proses geo-referensi citra dijitalnya.
    2. Mengolah citra dijital hingga lepas dari kesalahan.
    3. Menginterprestasikan citra.
    4. Dan memeriksa hasil interprestasi di lapangan.
c. Kelompok basisdata / *data automation* bertugas untuk:
    1. Mengkonversikan koordinat, peta hasil survei, dan hasil interpretasi citra dijital ke bentuk *layer* vektor.
    2. Merancang struktur tabel basisdata dan memasukkan atribut ke dalamnya.
    3. Mengintegrasikan (mendefiniskan relasi terkait) *layer-layer* dijital dengan layer dan tabel basis datanya (Prahasta, 2014).
d. Kelompok kartografi bertugas untuk:
    1. Mengumpulkan peta manuskrip beserta *layer* peta dijital.
    2. Memilih sistem proyeksi, simbol, warna, dan ukuran untuk setiap komponen petanya.
    3. Menyusunnya ke dalam komposisi peta (*layout*) yang siap saji baik di monitor dan di media sejenis kertas.
e. Kelompok jaringan bertugas untuk:
    1. Merancang topologi jaringan yang efisien.
    2. Mengimplementasikan jaringan komputer.

3. Menentukan kewenangan setiap kelompok di dalam sistem komputer yang bersangkutan.

f. Kelompok aplikasi bertugas untuk menganalisis kebutuhan pengguna (*system requirements* dan *software requirements*), merancang mengimplementasikan, membuat contoh data, menguji, dan memelihara sistem yang dikembangkan.

Setiap kelompok punya peran tersendiri selama periode tertentu. Seiring dengan berjalannya waktu dan proses belajar, setiap kelompok dapat bertukar peran pada pekerjaan lainnya. Meskipun demikian, karena contoh tugas-tugas kelompok pekerjaan di atas masih berada di dalam lingkup disiplin yang sama, maka setiap kelompok dapat melakukan tugas-tugas kelompok lainnya.

Sebagaimana terlihat bahwa mentransformasikan data menjadi informasi membutuhkan masukan lainnya, berupa ketentuan, pustaka, parameter, konstanta, kriteria, dan bahkan referensi. Sementara itu, setiap informasi yang dihasilkan akan disimpan di media (basis data) yang suatu saat siap dipanggil kembali (Prahasta, 2014).

(Husda, 2012) mengatakan, sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang–orang, fasilitas, teknologi, media prosedur–prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting, memproses tipe transaksi rutin tertentu, memberi sinyal, kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Informasi dalam suatu lingkungan sistem informasi harus mempunyai persyaratan umum sebagai berikut:

1. Harus diketahui oleh penerima sebagai referensi yang tepat.
2. Harus sesuai dengan kebutuhan yang ada dalam proses pembuatan / pengambilan keputusan.
3. Harus mempunyai nilai *surprise*, yaitu hal yang sudah diketahui hendaknya jangan diberikan.
4. Harus dapat menuntun pemakai untuk membuat keputusan. Suatu keputusan tidak selalu menuntut adanya tindakan.

Sistem informasi harus mempunyai beberapa sifat seperti:

1. Pemrosesan informasi yang efektif. Hal ini berhubungan dengan pengujian terhadap data yang masuk, pemakaian perangkat keras dan perangkat lunak yang sesuai.
2. Manajemen informasi yang efektif. Dengan kata lain, operasi manajemen, keamanan dan keutuhan data yang ada harus diperhatikan.
3. Keluwesan. Sistem informasi hendaknya cukup luwes untuk menangani suatu macam operasi.
4. Kepuasaan pemakai. Hal yang paling penting adalah pemakai mengetahui dan puas terhadap sistem informasi (Husda, 2012).

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), yaitu:

1. Komponen *input* atau komponen masukan.
2. Komponen model.

3. Komponen *output* atau komponen keluaran.
4. Komponen teknologi.
5. Komponen basis data.
6. Komponen kontrol atau komponen pengendalian.

(Husda, 2012) mengatakan, keenam komponen ini harus ada bersama – sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya, yaitu pengolahan data dan tidak dapat mencapai tujuannya, yaitu menghasilkan informasi yang relevan, tepat waktu dan akurat. Komponen–komponen dari sistem informasi ini dapat digambarkan sebagai berikut:

1. Blok Masukan (*Input Block*)

   Input merupakan data yang masuk ke dalam sistem informasi.

2. Blok Model (*Model Block*)

   Kombinasi prosedur, logika, dan model matemetik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

   Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

   Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

   Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak memanipulasinya.

6. Blok Kendali (*Control Block*)

   Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan–gangguan terhadap sistem.

Menurut (Husda, 2012), cara pandang pengklasifikasian sistem informasi adalah:

1. Level Organisasi

   Sistem informasi menurut level organisasi:

   a. Sistem informasi departemen

      Contoh: Sistem Informasi SDM (*HRIS*).

   b. Sistem informasi perusahaan (*enterprise information system*)

      Contoh: sistem informasi perguruan tinggi.

   c. Sistem informasi antar organisasi

      Contoh: *e-Commerce*.

2. Sistem Informasi Fungsional

   Sistem informasi berdasarkan area fungsional ditujukan untuk memberikan informasi bagi kelompok orang yang berada pada bagian tertentu dalam perusahaan.

   Contoh:

   a. Sistem informasi akuntansi

      Sistem informasi yang menyediakan informasi yang dipakai oleh fungsi akuntansi (departemen / bagian akuntansi).

   b. Sistem informasi keuangan

   c. Sistem informasi yang menyediakan informasi pada fungsi keuangan yang menyangkut keuangan perusahaan.

      Contoh: *Cash Flow* dan informasi pembayaran (Husda, 2012).

3. Klasifikasi Sistem Informasi Berdasarkan Fungsi

   a. Sistem Pemrosesan Transaksi (*Transaction Processing System / TPS*)

      Sebuah sistem yang meng-*capture* dan memproses data transaksi bisnis. Misalnya: pesanan, kartu absensi, pembayaran, KRS, reservasi, dll.

   b. Sistem Informasi Manajemen (*Management Information System / MIS*)

      Sistem informasi yang menyediakan pelaporan yang berorientasi manajemen berdasarkan pemrosesan transaksi dan operasi organisasi.

c. Sistem Pendukung Keputusan (*Decision Support System* / *DDS*)

Sistem informasi yang mengindentifikasi berbagai alternatif keputusan atau menyediakan informasi untuk membantu pembuatan keputusan.

d. Sistem Informasi Eksekutif (*Executive Information System* / *EIS*)

Sistem informasi yang diperuntukkan oleh manajer eksekutif untuk mendukung perencanaan bisnis dan menilai performa rencana tersebut.

e. Sistem Pakar (*Expert System*)

Sistem informasi yang meng-*capture* dan menghasilkan kembali pengetahuan asli pemecahan masalah atau para pengambil keputusan dan mensimulasikan kembali "pemikiran" ahli tersebut.

f. Sistem Komunikasi dan Kolaborasi (*Communication and Collaboration System*)

Sistem yang memungkinkan komunikasi lebih efektif antara orang–orang dalam maupun luar organisasi untuk meningkatkan kemampuan berkolaborasi.

g. Sistem Otomasi Kantor (*Office Automation System*)

Sistem informasi yang mendukung aktifitas bisnis kantor secara luas yang menyediakan aliran kerja yang diperbaiki antar personil (Husda, 2012).

4. Sistem Informasi Berdasarkan Dukungan Yang Tersedia
    a. Sistem Pemrosesan Transaksi (*Transaction Processing System* atau *TPS*)
    b. Sistem Informasi Manajemen (*Management Information System* atau *MIS*)
    c. Sistem Otomasi Perkantoran (*Office Automation System* atau *OAS*)
    d. Sistem Pendukung Keputusan (*Decision Support System* atau *DSS*)
    e. Sistem Informasi Eksekutif (*Executive Information System* atau *EIS*)
    f. Sistem Pendukung Kelompok (*Group Support System* atau *GSS*)
    g. Sistem Pendukung Cerdas (*Intelligent Support System* atau *ISS*) (Husda, 2012)

2.1.2. Pengertian Sistem Pendukung Keputusan

Menurut (Rohayani, 2013), definisi sistem pendukung keputusan yaitu sistem pendukung keputusan merupakan suatu sistem interaktif yang mendukung keputusan dalam proses pengambilan keputusan melalui alternatif–alternatif yang diperoleh dari hasil pengolaan data, informasi dan rancangan model.

Sistem pendukung keputusan dalam cakupan yang lebih sempit, yaitu sistem pendukung keputusan merupakan suatu sistem berbasis komputer yang interaktif,

yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah–masalah yang semi terstruktur.

Sistem pendukung keputusan adalah bagian dari sistem informasi berbasis komputer yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dari beberapa definisi tersebut, dapat diambil kesimpulan bahwa sistem pendukung keputusan untuk menyelesaikan masalah yang semi terstruktur, dengan memanfaatkan data yang ada kemudian diolah menjadi suatu informasi berupa usulan menuju suatu keputusan tertentu.

(Rohayani, 2013) mengatakan, karakteristik dari sistem pendukung keputusan yaitu:

1. Sistem pendukung keputusan memberikan dukungan bagi pengambil keputusan pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
2. Dukungan untuk semua level manajerial, dan eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk keputusan independen dan sekuensial.
5. Dukungan di semua fase proses pengambilan keputusan, yaitu *intelligence*, *design*, *choice*, dan *implementation*.
6. Dukungan di berbagai proses dan gaya yang berbeda-beda.
7. Adaptivitas sepanjang waktu.
8. Mudah untuk digunakan *user*.
9. Peningkatan efektivitas dari pengambilan keputusan daripada efisiensi.

10. Kontrol penuh oleh pengambil terhadap semua langkah proses pengambilan keputusan.
11. Pengguna akhir bisa mengembangkan dan memodifikasi sendiri sistem sederhana.
12. Biasanya, model-model digunakan untuk menganalisis situasi pengambilan keputusan.
13. Akses disediakan untuk berbagai sumber daya, *format*, dan tipe, mulai dari sistem informasi sampai sistem berorientasi objek.
14. Dapat digunakan sebagai *standalone* oleh seorang pengambil keputusan pada suatu lokasi atau didistribusikan di suatu organisasi secara keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.

Tujuan dari sistem pendukung keputusan adalah sebagai berikut:

1. Membantu manajer dalam pengambilan keputusan atas masalah semi-terstruktur.
2. Meningkatkan efektivitas keputusan yang diambil manajer lebih daripada perbaikan efesiensinya.
3. Peningkatan produktivitas. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berasal dari berbagai lokasi yang berbeda-beda (Rohayani, 2013).

2.1.3. Pengertian *SDLC*

Menurut (Rosa & Shalahuddin, 2011), *SDLC* atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* 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 *best practice* atau cara–cara sudah teruji dengan baik).

Tahapan–tahapan yang ada pada *SDLC* secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)

   Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

   Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*Planning*)

   Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (resources) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)

   Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

   Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi–fungsi yang dibutuhkan (Rosa & Shalahuddin, 2011).

6. Pengembangan (*development*)

   Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau file pengujian pengodean, pengompilasian, memperbaiki dan membersihkan program, peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

   Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (quality assurance) dan *user*. Menghasilkan laporan analisis penjualan.

8. Implementasi (*implementation*)

   Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian (Rosa & Shalahuddin, 2011).

9. Operasi dan pemeliharaan (*operations and maintenance*)

   Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

    Mendeskripsikan aktivitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktivitas *user*.

(Rosa & Shalahuddin, 2011) mengatakan, *SDLC* memiliki beberapa model dalam penerapan tahapan prosesnya, yaitu:

1. Model *Waterfall*

   Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).

   a. Analisis kebutuhan perangkat lunak

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

   b. Desain

      Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat

lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan (Rosa & Shalahuddin, 2011).

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

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

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan 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.

(Rosa & Shalahuddin, 2011) mengatakan, dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai alurnya karena sebab berikut:

a. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan. Pelanggan sering kali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan.

Dengan berbagai kelemahan yang dimiliki model air terjun tapi model ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikan model pengembangan perangkat lunak.

Model air terjun sangat cocok digunakan kebutuhan pelanggan sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari model air terjun adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan (tidak ada tumpang tindih pelaksanaan tahap).

2. Model *Prototype*

   Model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.

3. Model *Rapid Application Development* (*RAD*)

   Rapid Application Development (*RAD*) adalah model proses pengembangan perangkat lunak yang bersifat *incremental* terutama untuk waktu pengerjaan yang pendek. Model *RAD* adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak.

4. Model Iteratif

   Model iteratif (*iterative model*) mengkombinasikan proses-proses pada model air terjun dan iteratif pada model *prototype*. Model inkremental akan menghasilkan versi–versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (inkremen / *increment*).

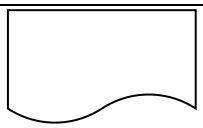
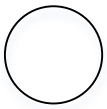
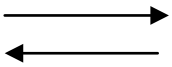
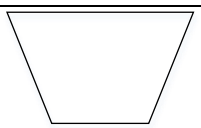
5. Model Spiral

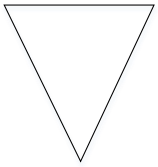
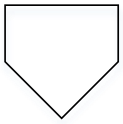
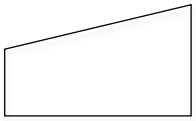
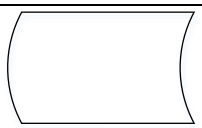
   Model Spiral (*spiral model*) memasangkan iteratif pada model *prototype* dengan kontrol dan aspek sistematik yang diambil dari model air terjun. Model spiral menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya (*increment*) (Rosa & Shalahuddin, 2011).

2.1.4. Aliran Sistem Informasi

Menurut (Ismael, 2017), aliran sistem informasi (ASI), merupakan bagan yang menunjukkan pekerjaan serta keseluruhan dari sistem. Bagian ini menjelaskan urutan dari prosedur-prosedur yang ada dalam sistem. Adapun simbol-simbol yang dapat digunakan pada aliran sistem informasi (ASI) ini adalah sebagai berikut:

Tabel 2. 1 Simbol Aliran Sistem Informasi

Simbol	**Deskripsi**
	Proses Komputer
	Dokumen
	Penghubung
	Garis alur
	Output di monitor
	Manual Proses

Simbol	Deskripsi
	*File Storage*
	Penghubung antar halaman
	*Input* menggunakan *keyboard*
	*Input output* menggunakan disket (Ismael, 2017)

2.1.5. *Unified Modeling Language*

Menurut (Rosa & Shalahuddin, 2011), *Unified Modeling Language* (*UML*), adalah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. *UML* muncul karena adanya kebutuhan pemodelan visual untuk mengspesifikasikan, menggambarkan, membangun, dan dokumentasi, dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks–teks pendukung.

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

Penulis menggunakan empat diagram *UML* yang akan penulis jelaskan pada berikut:

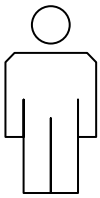
1. *Use Case Diagram*

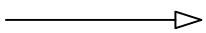
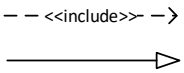
   Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang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

   Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa saja yang disebut aktor dan *use case* (Rosa & Shalahuddin, 2011).

Tabel 2. 2 Simbol *Use Case*

Simbol	**Deskripsi**
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di

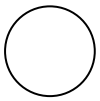
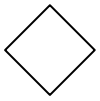
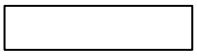
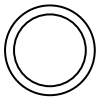
Simbol	Deskripsi
	awal frase nama *use case*.
Aktor / *actor*  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor (Rosa & Shalahuddin, 2011).
Asosiasi / *association* ———————	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi / *extend* – – <<extend>>– –>	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu, mirip dengan prinsip *inheritance* pada

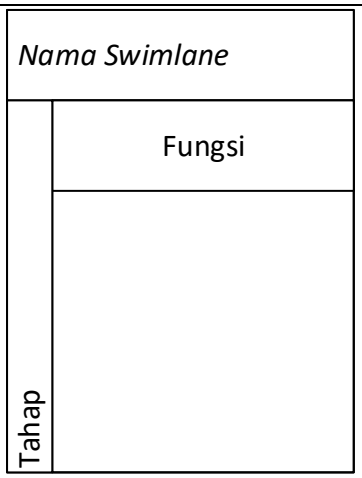
Simbol	Deskripsi
	pemrograman berorientasi objek, biasanya *use case* tambahan.
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / *include* 	Fungsinya atau sebagai syarat dijalankan *use case* ini.

2. *Activity Diagram*

Menurut (Rosa & Shalahuddin, 2011), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2. 3 Simbol *Activity Diagram*

Simbol	**Deskripsi**
Status Awal 	Status awak aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu (Rosa & Shalahuddin, 2011).
Penggabungan / *join* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Simbol	Deskripsi
	

3. *Class Diagram*

(Rosa & Shalahuddin, 2011) mengatakan, diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Atribut merupakan variabel–variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi–fungsi yang dimiliki oleh suatu kelas.

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

a. Kelas *main*

Kelas yang memiliki fungsi awal diekseskusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

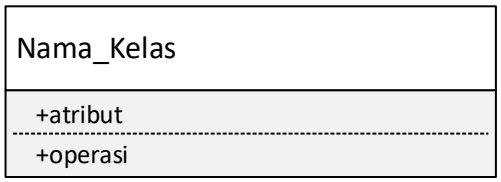
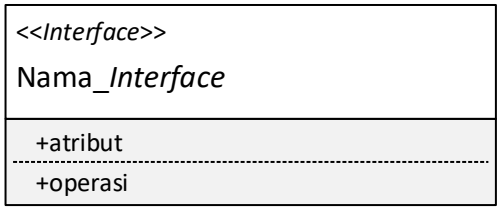
c. Kelas yang diambil dari pendefinisian *use case*

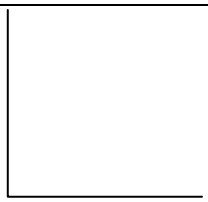
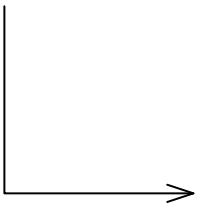
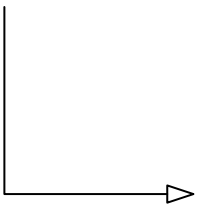
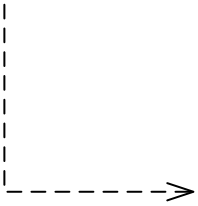
   Kelas yang menangani fungsi – fungsi yang harus ada diambil dari pendefinisian *use case*.

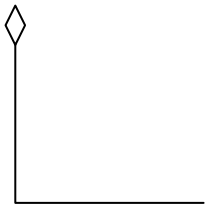
d. Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data (Rosa & Shalahuddin, 2011).

Tabel 2. 4 Simbol *Class Diagram*

Simbol	**Deskripsi**
Kelas 	Kelas pada struktur sistem
Antarmuka / *interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
Asosiasi / *association*	Relasi antara kelas dengan makna umum,

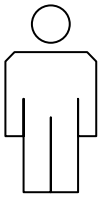
Simbol	Deskripsi
	asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah / *directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain (Rosa & Shalahuddin, 2011).
Generalisasi 	Relasi antar kelas dengan makna generalisasi–spesialisasi (umum-khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation*	Relasi antar kelas dengan makna semua

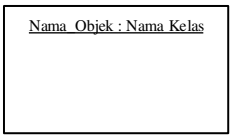
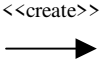
Simbol	Deskripsi
	bagian (*whole-part*).

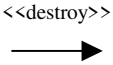
4. *Sequence Diagram*

(Rosa & Shalahuddin, 2011) mengatakan, diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode–metode yang dimiliki kelas yang diintansiasi menjadi objek itu.

Tabel 2. 5 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  Nama_Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan

Simbol	Deskripsi
	menggunakan kata benda di awal frase nama aktor.
Garis hidup / *lifeline*	Menyatakan kehidupan suatu obyek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan (Rosa & Shalahuddin, 2011).
Pesan tipe *Create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *call* 1 : nama_metode()	Menyatakan suatu objek

Simbol	Deskripsi
	memanggil operasi / metode yang ada pada obyek lain atau dirinya sendiri.
Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada obyek yang menerima kembalian.
Pesan tipe *destroy* 	Menyatakan suatu obyek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy* (Rosa & Shalahuddin, 2011).

2.2. Tinjauan Teori Khusus

Dalam bagian ini, peneliti akan menjelaskan mengenai teori-teori khusus menurut pendapat ahli.

2.2.1 Pengertian Pajangan

Pengertian pajangan atau *display* adalah keinginan membeli sesuatu, yang tidak didorong oleh seseorang, tetapi didorong oleh daya tarik, atau oleh pengliatan ataupun oleh perasaan lainnya (Potoh, Sylcyljeova Moniharapon, & Roring, 2017).

2.2.2 Pengertian *Data Mining*

(Pratama & Sihwi, 2014) mengatakan, *data mining* adalah proses yang menggunakan teknik statistik, matematika kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengindentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dan dari berbagai *database* besar.

Karakteristik *data mining* adalah sebagai berikut:

1. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu tidak diketahui sebelumnya.
2. *Data mining* biasa menggunakan data yang sangat besar. Biasanya data yang besar digunakan untuk membuat hasil lebih dipercaya.

3. *Data mining* berguna untuk membuat keputusan yang kritis, terutama dalam strategi (Pratama & Sihwi, 2014).

(Kusrini & Taufiq, 2009) mengatakan, istilah *data mining* dan *knowledge discovery in database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*. Proses KDD dijelaskan sebagai berikut.

1. *Data Selection*

   Pemilihan (seleksi) data dari sekempulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing / Cleaning*

   Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relavan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation*

   Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

   Data Mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat begantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation*

   Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

(Kusrini & Taufiq, 2009) mengatakan, *Data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

1. Deskripsi

   Terkadang peneliti dan analisis secara sederhana ingin mencoba cari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam

data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola kecendrungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan dara sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

   Contoh lain yaitu estimasi nilai indeks prestasi kumulatif mahasiswa program pascasarjana dengan melihat nilai indeks prestasi mahasiswa tersebut pada saat mengikuti program sarjana (Kusrini & Taufiq, 2009).

3. Prediksi

   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil aka nada di masa mendatang.

Contoh prediksi dalam bisnis dan penelitian adalah:

a. Prediksi harga beras dalam tiga bulan yang akan datang.
b. Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

Contoh lain klasifikasi dalam bisnis dan penelitian adalah:

a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa (Kusrini & Taufiq, 2009).

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain.

Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan benilai minimal.

Contoh pengklusteran dalam bisnis dan penelitian adalah:

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.
c. Melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah yang besar (Kusrini & Taufiq, 2009).

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

Contoh asosiasi dalam bisnis dan penelitian adalah:

a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.
b. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan (Kusrini & Taufiq, 2009).

2.2.3 Pengertian Algoritma Apriori

(Listriani, Setyaningrum, & Eka, 2016) menjelaskan, algoritma apriori adalah satu algoritma dasar yang diusulkan oleh *Agrawal* dan *Srikan* pada tahun 1994 untuk menemukan *frequent itemsets* pada aturan asosiasi *Boolean*. Ide utama pada algoritma apriori adalah mencari *frequent itemsets* (himpunan *item–item* yang memenuhi *support*) dari basis data transaksi, dan menghilangkan *itemset* dengan frekuensi yang rendah berdasarkan level minimum *support* yang telah ditentukan sebelumnya. Selanjutnya membangun aturan asosiasi dari *itemsets* yang memenuhi nilai minimum *confidence* dalam basis data.

Untuk membentuk kandidat *itemset* ada dua proses utama yang dilakukan algoritma apriori, yaitu:

1. *Join Step* (Penggabungan)

   Pada proses ini setiap *item* dikombinasikan dengan *item* lainnya sampai tidak terbentuk kombinasi lagi.

2. *Prune Step* (Pemangkasan)

   Pada proses ini, hasil dari item yang dikombinasikan tadi kemudian dipangkas dengan menggunakan minimum *support* yang telah ditentukan *user* (Listriani et al., 2016).

2.2.4 *Association Rules*

Menurut (Kusrini & Taufiq, 2009), *association rules* adalah teknik *data mining* yang berguna untuk menemukan suatu korelasi atau pola yang terpenting / menarik dari sekumpulan data besar. Metodologi dasar analisis asosiasi dibagi menjadi dua tahap:

1. Analisis pola frekuensi tinggi

   Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus sebagai berikut:

$$\boldsymbol{Support\ (A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Jumlah\ semua\ transaksi}}$$

Rumus 2. 1 Rumus Analisis Pola Frekuensi Tinggi Satu *Item*

   Sementara item, nilai *support* dari dua *item* diperoleh dari rumus kedua berikut:

$$\boldsymbol{Support\ (A,B) = \frac{Jumlah\ transaksi\ mengandung\ A\ dan\ B}{Jumlah\ semua\ transaksi}}$$

Rumus 2. 2 Rumus Analisis Pola Frekuensi Tinggi Dua *Item*

Untuk mencari nilai *support item* A ataupun *item* A dan *item* B, maka total semua transaksi di suatu periode yang mengandung *item* A atau *item* A dan *item* B dibagi dengan total semua transaksi di suatu periode.

2. Pembentukan aturan asosiatif

   Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* aturan asosiatif A > B. Nilai *confidence* dari aturan A > B diperoleh dari rumus berikut:

$$\boldsymbol{Confidence(A,B) = \frac{Jumlah\ transaksi\ mengandung\ A\ dan\ B}{Jumlah\ transaksi\ mengandung\ A}}$$

Rumus 2. 3 Rumus Pembentukan Aturan Asosiatif

   Untuk mencari nilai *confidence item* A dan B, maka total semua transaksi di suatu periode yang mengandung *item* A dan *item* B dibagi dengan jumlah transaksi di suatu periode yang mengandung item A.

Seorang analisis mungkin hanya akan mengambil aturan yang memiliki *support* dan *confidence* yang tinggi. Aturan yang kuat adalah aturan-aturan yang melebihi kriteria *support* dan *confidence* minimum. Misalnya seorang analisis menginginkan aturan yang memiliki *support* lebih dari 20% dan *confidence* lebih dari 35% (Kusrini & Taufiq, 2009).

2.2.5 Pengertian Aplikasi *Web*

(Arif, Wanda, & Abdi, 2013) mengatakan, pada awalnya aplikasi *web* dibangun dengan hanya menggunakan bahasa yang disebut *HTML* (*HyperText Markup Language*). Pada pengembangan berikutnya, sejumlah skrip dan objek

dikembangkan untuk memperluas kemampuan *HTML* seperti *PHP* dan *ASP* pada skrip dan *Appalet* pada objek. Aplikasi *web* dapat dibagi menjadi dua jenis, yaitu aplikasi *web* statis, dan dinamis.

Web statis dibentuk dengan menggunakan *HTML*. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi *web* dinamis. Pada aplikasi *web* dinamis, perubahan informasi dalam halaman *web* dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi *web* dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab *webmaster*.

Arsitektur aplikasi *web* meliputi klien, *web server*, *middleware*, dan basis data. Klien berinteraksi dengan *web server*. Secara internal, *web server* berkomunikasi dengan *middleware*, dan *middleware* berkomunikasi dengan basis data. Contoh *middleware* adalah *PHP* dan *ASP*. Pada mekanisme aplikasi *web* dinamis, terjadi penambahan proses yaitu *server* menerjemahkan kode *PHP* menjadi kode *HTML*. Kode *PHP* akan diterjemahkan oleh mesin *PHP* yang akan diterima oleh klien (Arif *et al*., 2013).

Berikut adalah pengertian beberapa bahasa pemrograman untuk pembuatan aplikasi *web* yang penulis gunakan:

2.2.5.1. *HTML*

Pengertian *HTML* adalah bahasa pemrograman standar yang dipakai untuk menampilkan informasi pada halaman *web*. *HTML* menampilkan informasi dalam bentuk *hypertext* dan sesuai dengan namanya. Bahasa ini menggunakan tanda (*markup*) disebut dengan *tag* untuk menandai perintah-perintahnya (Jamaludin & Sugiarto, 2017).

(Sianipar, 2015) menjelaskan versi *HTML* yang terbaru adalah *HTML5*. *HTML5* adalah sebuah *markup* untuk menstrukturkan dan menampilkan isi dari halaman *web*. *HTML* (yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, *HTML4*, pada tahun 1997) dan hingga bulan Juni 2011 tetap dalam proses pengembangan. Tujuan utama pengembangan *HTML5* adalah untuk memperbaiki teknologi *HTML* agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya *World Wide Web Consortium*, *W3C* untuk mendefinisikan sebuah bahasa markup tunggal yang dapat ditulis dengan cara *HTML* ataupun *XHTML*. *HTML5* merupakan jawaban atas pengembangan *HTML* 4.01 dan *XHTML* 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat *web*.

2.2.5.2. *CSS*

Menurut (Jamaludin & Sugiarto, 2017), *Cascading Style Sheet* secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan

tag HTML seperti *font*, *color*, *text*, *table*, menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan.

Menurut (Sianipar, 2015), *Cascading Style Sheet* (*CSS*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. *CSS* bukan merupakan bahasa pemrograman. *CSS* dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antarparagraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. *CSS* adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya *CSS* pengguna dapat menampilkan halaman yang sama dengan format yang berbeda.

2.2.5.3. *PHP*

(Jamaludin & Sugiarto, 2017) mengatakan, *PHP* merupakan kependekan dari *Hypertext PreProcessor*. *PHP* tergolong sebagai perangkat lunak *open source* yang diatur dalam aturan *General Purpose Licence* (*GPL*). Pemrograman *PHP* sangat cocok dikembangkan dalam lingkungan *web*, karena *PHP* bisa dilekatkan pada *script HTML* atau sebaliknya. *PHP* tergolong juga sebagai bahasa pemrograman yang berbasis *server* (*server side scripting*) (Jamaludin & Sugiarto, 2017).

Menurut (Hariyanto, 2017), *PHP* merupakan kependekkan dari *Personal Home Page* (Situs Personal). *PHP* pertama kali dibuat oleh *Rasmus Lerdorf* pada tahun 1995. Pada waktu itu, *PHP* masih bernama *Form Interpreted* (*FI*), yang

wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah dan formulir dari *web*.

PHP dapat digunakan secara gratis (*free*) dan bersifat *Open Source*. *PHP* dirilis dalam lisensi *PHP Licence*, sedikit berbeda dengan lisensi *GNU General Public Licence* (*GPL*) yang biasa digunakan untuk proyek *Open Source*.

Versi terbaru dari bahasa pemrograman *PHP* adalah versi 5.6.4 yang resmi dirilis pada tanggal 18 Desember 2014. Pada awalnya, *PHP* hanya disebut sebagai *Personal Home Page*. Sesuai dengan namanya, *PHP* digunakan untuk membuat *website* pribadi. Dalam perkembangannya, *PHP* menjelma menjadi bahasa pemrograman *web* yang *powerful* dan tidak hanya digunakan untuk membuat halaman *web* sederhana.

Hingga saat ini, *PHP* digunakan pada banyak aplikasi berbasis *web*. Beberapa aplikasi berbasis *web* menggunakan *PHP* yang cukup *familiar* yaitu *Facebook*, *Twitter*, *Blogspot*, *Ebay*, *Wikipedia*, *Google*, *Google Map*, *Youtube*, *Instagram*, *Likedin*, *Amazon*, Kaskus, Detik.com, dan lain-lain (Hariyanto, 2017).

PHP disebut bahasa pemrograman *server-side*, karena *PHP* di proses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser* (*client*).

Intinya, *PHP* adalah sebuah bahasa pemrograman yang bisa membantu kita dalam membuat aplikasi apa saja dan bisa diakses oleh siapa saja dengan menggunakan teknologi *server-side*. *Server-side* yang dimaksud disini adalah *PHP* hanya akan berjalan pada aplikasi berbasis *server*, baik ini *server* yang berjalan di komputer *local* (*localhost*) maupun *server* yang berjalan secara *online*.

Oleh karena *PHP* berjalan pada *server*, maka *PHP* tidak bisa berjalan ketika diakses secara langsung.

Dalam pemrograman, tentunya *PHP* bukanlah satu-satunya bahasa pemrograman web. Selain *PHP*, kita juga mengenal *ASP* (*Active Server Pages*), *JSP* (*Java Server Page*), *Perl*, *Ruby*, *Cold Fusion*, dan lain-lain. Tentu ada banyak alasan mengapa memilih *PHP*.

(Hariyanto, 2017) juga mengatakan, ada beberapa kelebihan *PHP* dibandingkan bahasa pemrograman *web* lainnya, diantaranya adalah:

1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung *PHP* dapat ditemukan dimana-mana, dari mulai *Apache*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. *PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux*, *Unix*, *Macintosh*, *Windows*) dan dapat dijalankan secara secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

Selain itu, ada banyak kelebihan lain dari *PHP* ini, diantaranya:

1. Mampu terkoneksi dengan banyak pilihan *database* seperti *MySQL*, *ORACLE*, *PostgresSQL*, *dBase*, *IBM*, *DB2*, *ODBC*, *Ms. Access*, *SQL Lite*, dan *database* lainnya.
2. Gratis, karena *PHP* bersifat *open source*, sehingga kita menggunakannya kapan saja dan dimana saja tanpa harus bayar.
3. Komponen *PHP* sangat handal dan ringan. Ibaratnya tenaga sekuat gajah, tetapi konsumsi BBM nya sedikit semut. Perintah-perintah dalam *PHP* sangat mumpuni dan dapat dijalankan tanpa membenahi *server* terlalu berat dan cepat.
4. Dari banyak hal di atas, yang paling penting *PHP* itu ternyata adalah bahasa pemrograman yang paling banyak digunakan dalam pengembangan aplikasi-aplikasi berbasis *internet*.

Mengacu pada data survei yang dilakukan per tanggal 1 Juli 2016, *PHP* dinobatkan sebagai *most popular server-side programming languages*. Artinya *PHP* adalah bahasa pemrograman *server-side* paling popular yang digunakan lebih dari 82% *website* yang ada (Hariyanto, 2017).

2.2.6 Pengertian *Adobe Dreamweaver CS 6*

(Jamaludin & Sugiarto, 2017) mengatakan, *Adobe Dreamweaver* adalah perangkat lunak untuk mendesain *web* yang menyediakan kemampuan visual yang intuitif termasuk pada tingkat kode, yang dapat digunakan untuk membuat dan meng*edit website HTML* serta aplikasi *mobile* seperti *smartphone*, *tablet*, dan perangkat lainnya.

Adobe Dreamweaver dikembangkan oleh perusahaan *Software Adobe Inc*. saat ini versi yang beredar adalah *Adobe Dreamweaver CS 6* yang memiliki banyak fitur, antara lain:

1. *Layout Fluid Grid*
2. Peningkatan *support jQuery Mobile*
3. Transisi *CSS3* dan *HTML5*
4. Fitur *Live View* diperbarui
5. Mendukung integrasi *CMS*
6. Isyarat kode yang lebih spesifik (Jamaludin & Sugiarto, 2017).

2.2.7 Pengertian Basis Data

Menurut (Jamaludin & Sugiarto, 2017), basis data terdiri dari dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai berkas atau gudang, tempat bersarang / berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili sebuah objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data sendiri dapat didefinisikan sebagai kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Berikut aplikasi basis data yang penulis gunakan yaitu:

2.2.7.1. *MySQL*

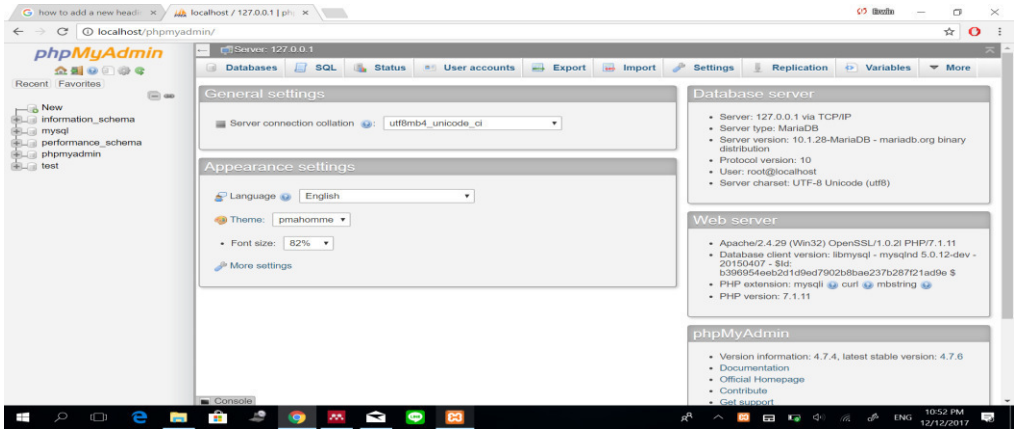
MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL DBMS* yang *multithread* dan *multiuser*, dengan sekitar enam juta instalasi di seluruh dunia. *MySQL* biasanya digunakan untuk aplikasi berbasis *web* dan dikombinasikan dengan bahasa pemrograman berbasis *web*, yaitu *PHP* (Jamaludin & Sugiarto, 2017).

(Enterprise, 2017) mengatakan, *MySQL* adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak sepakat dipakai untuk berbagai kebutuhan. *MySQL* dikembangkan oleh *MySQL AB Swedia*. Hampir sebagian besar aplikasi *website* yang ada di *internet* dikembangkan menggunakan *MySQL* dan bahasa pemrograman lainnya, seperti *PHP*.

Berikut ini hal-hal yang menyebabkan *MySQL* menjadi begitu populer:

1. Berlisensi *open-source*, sehingga dapat menggunakan secara gratis.
2. Merupakan program yang *powerful* dan menyediakan fitur yang lengkap.
3. Menggunakan bentuk standar bahasa data *SQL*.
4. Dapat bekerja dengan banyak sistem operasi dan dengan bahasa pemrograman, seperti *PHP*, *PERL*, *C*, *C++*, *JAVA*, dan lain-lain.
5. Bekerja cepat dan baik, bahkan dengan data *set* yang banyak.
6. Sangat mudah digunakan dengan *PHP* untuk pengembangan aplikasi *web*.
7. Mendukung banyak *database*, sampai 50 juta baris atau lebih dalam suatu tabel.
8. Dapat dikustomisasi sesuai dengan keinginan.

2.2.7.2. *PhpMyadmin*



Gambar 2. 1 *phpMyAdmin*

phpMyAdmin adalah aplikasi *web* yang dibuat oleh *phpMyAdmin.net* yang digunakan untuk administrasi *database MySQL* (Jamaludin & Sugiarto, 2017).

2.2.7.3. *XAMPP*

XAMPP adalah suatu paket yang berisi kumpulan *software* (*Apache*, *MySQL*, *PHP*) yang digunakan untuk dan dapat berjalan pada *platform* yang berbeda. *XAMPP* dikembangkan oleh *Apache Friends*, sebuah organisasi nirlaba yang bekerja pada proyek-proyek pengembangan sistem (Jamaludin & Sugiarto, 2017).

2.3. Penelitian Terdahulu

Berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan judul penelitan yang penulis angkat:

Tabel 2. 6 Tabel Penelitian Terdahulu

Tahun	Peneliti	Judul	Jurnal	Hasil
2013	(Bansal & Bhambhu)	*Execution of APRIORI Algorithm of Data Mining Directed Towards Tumultuous Crimes Concerning Women*	Vol. 3 No. 9 ISSN : 2277 128X	Penelitian ini menggunakan aturan asosiasi untuk mencari kejahatan yang sering terjadi dan menggunakan algoritma apriori untuk menggali data tentang kejahatan terhadap wanita, dan menggunakan aplikasi *WEKA* untuk memperlihatkan hasilnya. *Dataset* yang dikumpulkan berasal dari *UCI repository*, dan juga didapatkan dari persidangan kejahatan terhadap wanita. Tujuan utama ini untuk mengecek data yang didapatkan dari *UCI*

Tahun	Peneliti	Judul	Jurnal	Hasil
				repository, dan menggunakan apriori terhadap data faktual kejahatan terhadap wanita untuk mengambil informasi yang menunjukkan rentang usia pelaku dan dimana untuk mencari pelaku. Pada bagian terakhir adalah perbandingan antara apriori dan apriori prediktif yang menunjukkan bahwa apriori lebih cepat dan bagus daripada apriori prediktif.
2014	(Pratama & Sihwi)	Penerapan *Association Rule* Apriori dalam Aplikasi *Business Analytic* Terhadap	Vol. 3 No. 2 ISSN : 2301 - 7201	Hasil yang diperoleh dari penelitian ini, berupa aplikasi yang menampilkan kualitas kelulusan dalam bentuk

Tahun	Peneliti	Judul	Jurnal	Hasil
		Data Kelulusan di Universitas Sebelas Maret(UNS)		*chart* yang mudah dipahami. Sehingga, dengan menggunakan penerapan *association rule* apriori dalam aplikasi *business analytic* dapat menghasilkan pengetahuan yang bisa digunakan dalam pengambilan keputusan.
2014	(Wulandari & Rahayu)	Pemanfaatan Algoritma Apriori Untuk Perancangan Tata Letak Barang di Toko Busana	ISSN : 1907 - 5022	Hasil penelitian berupa sistem berbasis *web* yang menghasilkan aturan asosiasi sehingga penataan ulang bisa dilakukan secara dinamis. Pengujian dengan dataset baru dan perbandingan dengan *WEKA* membuktikan bahwa sistem berhasil menerapkan algoritma

Tahun	Peneliti	Judul	Jurnal	Hasil
				apriori. Keberhasilan dinilai berdasarkan kesamaan aturan asosiasi pada *WEKA* dengan aturan asosiasi pada sistem. Penataan ulang berdasarkan data transaksi selama bulan Juli 2012 dengan minimum *support* 3% dan *confidence* 20% menunjukkan perlunya perubahan tata letak barang yang termasuk dalam kategori dalaman Jilbab.
2014	(Tanna & Ghodasara)	*Using Apriori with WEKA for Frequent Pattern Mining*	Vol. 12 No. 3 ISSN : 2231 - 5381	Penelitian ini menunjukkan bagaimana cara pendekatan berbeda bisa mencapai *frequent mining* beserta kesulitan-kesulitan yang akan

Tahun	Peneliti	Judul	Jurnal	Hasil
				dihadapi pada saat melakukan penelitian ini. Penelitian ini menunjukkan bagaimana cara menggunakan aplikasi *WEKA* untuk mencari aturan asosiasi dengan menggunakan algoritma apriori.
2014	(Adewole, Akintola, & Ajiboye)	*Frequent Pattern and Association Rule Mining from Inventory Database Using Apriori Algorithm*	Vol 7. No. 3 ISSN : 2006 - 1781	Penelitian ini bertujuan untuk mencari aturan asosiasi mengenai beberapa *item* di *database* transaksi penjualan yang besar untuk mengetahui pola belanja konsumen agar bisa meningkatkan penjualan. Algoritma apriori digunakan untuk mencari aturan asosiasi dari *database* tersebut.

Tahun	Peneliti	Judul	Jurnal	Hasil
				Penelitian ini menemukan bahwa ada banyak *item* yang dibeli secara berulang–ulang, dan apriori merupakan cara yang tepat untuk mencari aturan asosiasi dari kasus tersebut. Algoritma ini dibuat dengan menggunakan *PHP*, dan *MySQL* sebagai *database*nya. Algoritma ini menghasilkan *frequent itemset* dan aturan asosiasi yang kuat.
2015	(Tambun & Anofrizen)	Rancang Bangun Aplikasi *Data Mining* Untuk Menampilkan Informasi Tingkat Kelulusan	Vol. 1 No. 1 ISSN : 2460 - 8181	Dengan memanfaatkan data induk mahasiswa dan data kelulusan mahasiswa, diharapkan dapat menghasilkan informasi tentang tingkat

Tahun	Peneliti	Judul	Jurnal	Hasil
		Mahasiswa Dengan Algoritma Apriori		kelulusan dengan data induk mahasiswa melalui teknik *data mining*. Kategori tingkat kelulusan diukur dari lama studi dan IPK. Algoritma yang digunakan adalah algoritma apriori, informasi yang ditampilkan berupa nilai *support* dan *confidence* dari masing–masing tingkat kategori kelulusan.
2015	(Prananingrum *et al.*)	Aplikasi *Data Mining* Untuk Menampilkan Informasi Unggulan Produk Kerajinan Tangan.	Vol. 1 ISSN : 2460 - 4690	Pada penelitian ini dibuat suatu aplikasi *data mining* dengan algoritma apriori guna untuk mengelola data kerajinan tangan yang ada, dan juga menganalisis

Tahun	Peneliti	Judul	Jurnal	Hasil
				kemungkinan pertumbuhan produk-produk kerajinan tangan yang baru yang lebih unggul dengan melakukan klasifikasi dari data–data kerajinan yang sudah ditemukan.
2015	(Marthasari, Azhar, & Puspitaningrum)	Sistem Rekomendasi Penyewaan Perlengkapan Pesta Menggunakan *Collaborative Filtering* Dan Penggalian Aturan Asosiasi	Vol. 5 No. 1 ISSN : 2088-2130	Sistem ini memudahkan konsumen menentukan barang untuk dibeli dengan cara menampilkan produk yang terkait dengan salah satu produk yang dibeli atau dilihat konsumen. Salah satu teknik yang dapat digunakan adalah penggalian aturan asosiasi menggunakan algoritma apriori. Pada penelitian ini, dibuat

Tahun	Peneliti	Judul	Jurnal	Hasil
				sebuah *website* persewaan alat–alat pesta dengan menerapkan sistem rekomendasi. Sistem rekomendasi dibangun menggunakan aturan–aturan yang dihasilkan oleh algoritma apriori. Untuk dapat menampilkan barang rekomendasi digunakan nilai *support* 20, sedangkan nilai *confidence* digunakan untuk menentukan N-teratas barang untuk direkomendasikan.
2015	(Meilani & Utomo)	Aplikasi *Data Mining* Untuk Pola Permintaan Darah di UDD (Unit Donor Darah) PMI	ISBN : 978 – 602 – 70604 – 1 - 8	Dengan menghasilkan pola permintaan darah menggunakan algoritma apriori, maka dapat disimpulkan dari hasil uji

Tahun	Peneliti	Judul	Jurnal	Hasil
		Kota Surabaya Menggunakan Metode Apriori		yang telah dilakukan adalah penentuan minimum *support* akan memberikan hasil pola permintaan darah yang berbeda–beda sesuai dengan nilai *support*nya, dengan mempertimbangkan juga nilai *confidence*. Kombinasi pola *itemset* yang memiliki nilai *support* dan *confidence* lebih besar dari pola lainnya dapat dikatakan sebagai pola permintaan darah yang paling baik dan sebagai bahan pertimbangan dalam pengambilan keputusan. Pola permintaan darah yang terjadi di PMI pada

Tahun	Peneliti	Judul	Jurnal	Hasil
				kondisi tidak menghasilkan pola pada minimum *support* 67%.
2016	(Hutahaean *et al*.)	Analisa Dan Perancangan Aplikasi Algoritma Untuk Korelasi Penjualan Produk (Studi Kasus: Apotik Diory Farma)	Vol. 1 No. 1 ISSN : 2541 - 3724	Penelitian ini menggunakan proses *data mining* dengan metode *association rules* dan apriori. *Association rule* merupakan perhitungan *support* dan *confidence* dari suatu kombinasi *item*, sedangkan apriori merupakan pengambilan data dengan aturan asosiatif (*association rules*).
2016	(Listriani *et al.*)	Penerapan Metode Asosiasi Menggunakan Algoritma Apriori Pada Aplikasi	Vol. 9 No. 2 ISSN : 1979 - 9160	Hasil penelitian ini berupa aplikasi untuk menganalisa pola belanja yang mana pola yang dihasilkan dapat

Tahun	Peneliti	Judul	Jurnal	Hasil
		Analisa Pola Belanja Konsumen		dijadikan rekomendasi dalam menentukan strategi penjualan oleh pihak Gramedia.
2017	(Yasmiati, Wahyudi, & Susilo)	Pengembangan Aplikasi *Data Mining* Dengan Algoritma C4.5 dan Apriori di Fakultas Teknologi Informatika Universitas Respati Indonesia	Vol. 9 No. 1 ISSN : 2085 - 1669	Data nilai mahasiswa dan data hasil evaluasi belum pernah dianalisis lebih mendalam sehingga data tersebut digunakan dengan penerapan menggunakan algoritma C4.5 dan apriori dari metode *data mining*. Algoritma C4.5 untuk melakukan klasifikasi dengan hasil pembentukkan pohon keputusan (*decision tree*). Sedangkan algoritma apriori digunakan dengan cara pengkombinasian *item*

Tahun	Peneliti	Judul	Jurnal	Hasil
				dengan *item* yang lainnya dan menghasilkan nilai *support* dan *confidence*.

Sumber:

(Adewole *et al*., 2014),

(Bansal & Bhambhu, 2013),

(Hutahaean *et al*., 2016),

(Listriani *et al*., 2016),

(Marthasari *et al*., 2015),

(Meilani & Utomo, 2015),

(Prananingrum *et al*., 2015),

(Pratama & Sihwi, 2014),

(Tambun & Anofrizen, 2015),

(Tanna & Ghodasara, 2014),

(Wulandari & Rahayu, 2014),

(Yasmiati *et al*., 2017).