BAB II
TINJAUAN PUSTAKA

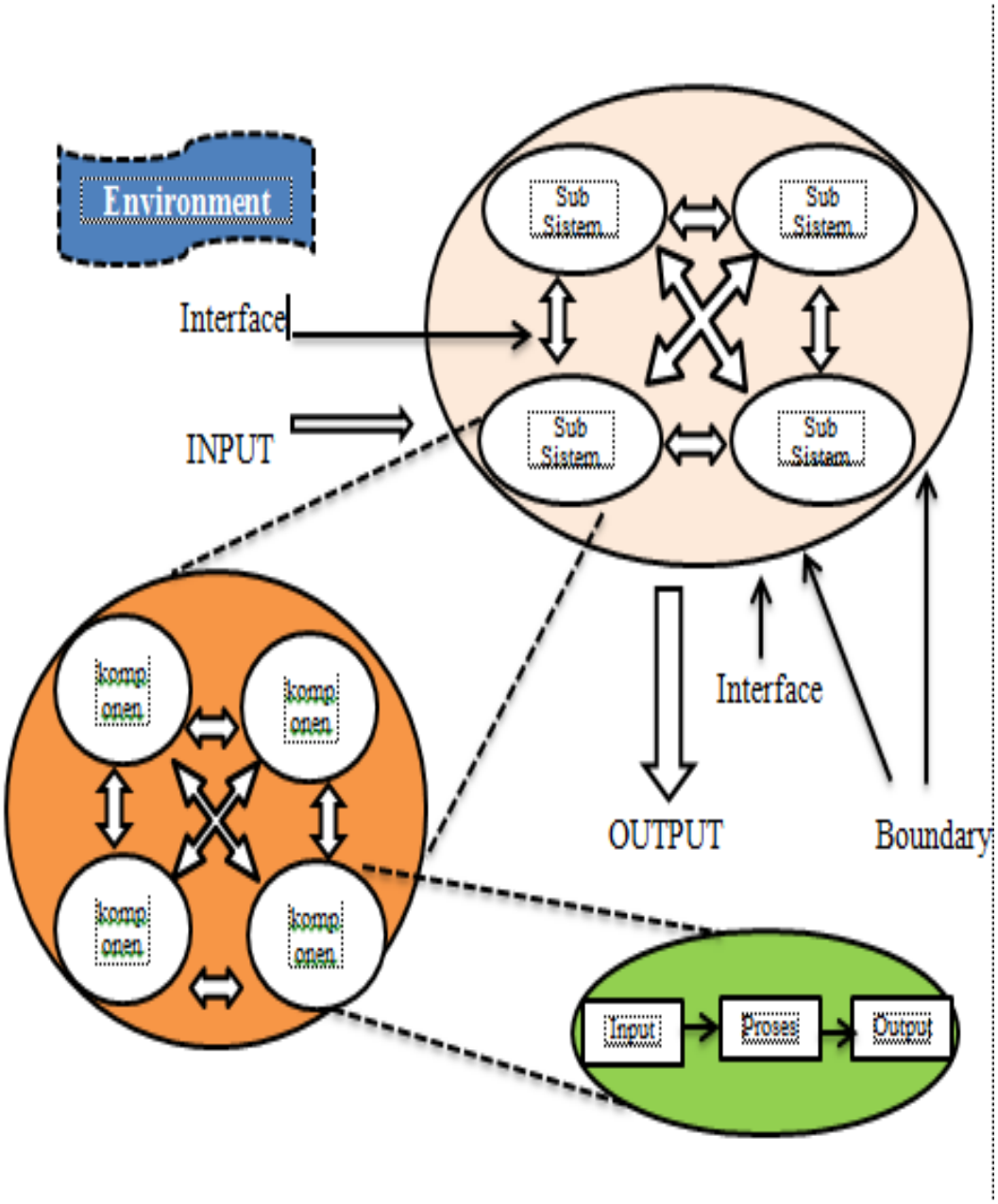
2.1 Tinjauan Teori Umum

Pada penulisan skripsi ini, penulis mengembangkan pemikiran lewat teori umum dan teori khusus yang menjadi dasar dan patokan toeri umum dan teori khusus dalam penulisan skripsi ini digunakan sebagai bahan acuan. Teori adalah seperangkat konsep, defenisi dan proposisi yang tersusun secara sistematis sehingga dapat digunakan untuk menjelaskan dan meramalkan fenomena (Sugiyono, 2012 : 52).

2.1.1 Sistem

Sistem (*system*) adalah kumpulan dari sub-sub sistem, elemen-elemen, prosedur-prosedur, yang saling berintegrasi untuk mencapai tujuan tertentu, seperti informasi, target atau goal. Karakter suatu sistem terdiri dari : komponen (*Components*), Batas Sistem (*Boundary*), Lingkungan luar Sistem (*Environments*), Penghubung (*Interface*), *input*, *process* dan *output*, Sasaran (*Objectives*), Tujuan (*Goal*). Menurut buku *Conceptual, Structure and Development*, "Sistem dapat bersifat abstrak atau fisis. Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsep yang saling tergantung" (Ali & Wangdra, 2010 : 8).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu (Ermatita, 2016).



Gambar 2. 1 Karakteristik Sistem

2.1.2 Karakteristik Sistem

Menurut (Muslihudin & Oktafianto, 2016 : 4-5) sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut dapat dikatakan sebagai suatu sistem. Suatu sistem memiliki katrakteristik atau sifat tertentu,yaitu:

1. Komponen Sistem

   Suatu sistem yang terdiri atas bagian-bagian yang saling berkaitan dan bervariasi yang bersama-sama mencapai beberapa sasaran. Sebuah sistem bukanlah seperangkat unsur yang tersusun secara teratur, tetapi terdiri atas unsur yang dapat dikenal dan saling melengkapi karena suatu maksud, tujuan dan sasaran.

2. Batas Sistem

   Batas Sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu ksatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar (*environment*)

   Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energy dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar

yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan menggangu kelangsungan hidup sistem.

4. Penghubung sistem (*interface*)

   Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran *(output*) untuk subsistem yang lainnya dengan melalui penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*input*)

   Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*output*)

   Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keuaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Missal nya untuk sistem komputer, panas yang dihasilkan adalah keluaran

yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. Sasaran Sistem (*goal*)

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak aka nada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.1.3 Informasi

Informasi (*information)* adalah data yang telah diolah dan dikembangkan menjadi suatu bentuk yang penting bagi sipenerima atau pengguna dan mempunyai nilai yang nyata dan konkrit atau dapat dirasakan mamfaatnya dalam keputusan-keputusan yang digunakan dimasa yang akan datang. *Output* informasi dari komputer digunakan oleh para Meneger, non Meneger atau pegguna keptusan, serta orang-orang dan organisasi-organisasi dalam lingkungan perusahaan. Yang lebih penting adalah pengguna informsi perlu mengerti informasi dan tujuan informasi itu (*Information Literate*) seperti informasi apa yang mereka butuhkan sesuai dengan kenutuhan pengguna, untuk apa informasi tersebut digunakan serta bagaimana mutu dan kualitas informasi yang dapat membantu dalam mengambil keputusan mereka mengindentifikasi dan memecahkan masalah, untuk memutuskan tindakan yang akan diambil saat ini maupun yang akan datang. (Ali & Wangdra, 2010 : 10).

Menurut (Winarno, 2006 : 6) informasi adalah data yang sudah diolah dan memiliki tujuan yang pasti sehingga berguna untuk pembuatan keputusan dan berguna bagi pengguna informasi terebut. Data adalah representasi atau gambaran suatu objek.

Gambar 2. 2 Data diolah menjadi Informasi

Gambaran dari kualitas dari suatu sistem informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat waktunya (*timelines*) dan relevan (*relevance*). John Burch dan Gary Grudnitski menggambarkan kualitas dari informasi denganbentuk bangunan yang ditunjang oleh tiga buah pilar.

1. Akurat, berarti informasi harus bebas dari kesalahn-kesalahan dan tidak bisa menyesatkan. Akurat juga berarti informasi harus jeals mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusk informasi tersebut
2. Tepat waktu, berarti informasi yang dating pada penerima tidak boleh terlambat. Informasi yang suadah using tidak akan mempunyai nilai lai. Karena informasi merupakan landasan di dalam pengambilan keputusan.
3. Relevan, berarti informasi tersebut mempunyai mamfaat untuk pemakainya. Relevansi informasi untuk tiap-tipa orang satu denagan yang lainnya berbeda. Misalnya informasi mengenai sebab-musabab kerusakan mesin

produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila di tujukan kepada ahli teknik perusahaan.

2.1.4 Sistem Informasi

Sistem informasi (*information system*) merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran sistem informasi. Dalam hal ini, TI merupakan salah satu komponen dalam perusahaan. Komponen-komponen yang lainnya adalah prosedur, struktur organisasi, sumberdaya manusia, produk, pelanggan, rekanan dan sebagainya.(Ali & Wangdra, 2010 : 13) .

Menurut (Tantra, 2012) Sistem Informasi adalah cara yang terorganisir untuk mengumpulkan, memasukkan, dan memproses data dan menyimpannya, mengelola, mengontrol dan melaporkannya sehingga dapat mendukung perusahaan atau organisasi untuk mencapai tujuan. Sistem informasi dapat bersifat formal dan informal, kedua perbedaan ini dapat dijelaskan sepeti yang akan digambarkan, sistem informasi yang bersifat formal memang secara resmi memiliki tanggung jawab dan peranan untuk menghasilkan informasi yang akurat, sedangkan sistem informasi informal adalah kebalikan dari system informasi formal, yang berasal dari bagian-bagian organisasi baik itu internal organisasi ataupun external organisasi yang tidak secara resmi memberikan informasi, seperti misalnya bagian legal.

2.1.5 Komponen Sistem Informasi

(Muslihudin & Oktafianto, 2016: 14) John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*controls block*).

1. Blok masukan (*input block)*

   Input mewakili data yang masuk ke daam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan,yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prosedur,logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajeman semua pemakai sistem.

4. Blok teknologi (*technology block*)

   Teknologi merupakan “kotak alat” (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, minyimpan dan mengakses data, menghasilkan danmengirimkan keluaran

dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*human atau brainware*), perangkat lunak (*sofware*) dan perangkat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beropeasi.

5. Blok basis data (*database block*)

   Basis data (*database)* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya,tesimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

6. Blok kendali (*controls block*)

   Banyak hal yang dapat merusk sistem informasi, seperti misalnya bencana alam, apai, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.1.6 *Internet*

Internet adalah sebagai jaringan komputer yang sangat luas dan besar dan mendunia, menghubungkan pemakai jaringan komputer dari satu negara ke negara lain di seluruh dunia sehingga memiliki koneksi yang saling terhubung menjadi satu kesatuan untuk mencapai tujuan, dimana di dalamnya terdapat berbagai sumber informasi dan fasilitas–fasilitas layanan internet yaitudiantaranya(Hastanti, Eka, Indah, & Wardati, 2015):

1. *Browsing* atau *surfing* yaitu kegiatan *"berselancar*" di internet, kegiatan ini seperti layaknya berjalan-jalan
2. *Elektronik mail (E-mail)* fasilitas ini digunakan untuk berkirim surat dengan orang lain, tanpa mengenal batas, waktu, ruang bahkan birokrasi. *searching* yaitu kegiatan mencari dan menemukan data atau informasi tertentu diinternet sesuai dengan kebutuhan pengguna surat elektronik.
3. *Catting* fasilitas ini digunakan untuk berkomunikasi secara langsung dengan orang lain di Internet sehingga mempermudah pengguan dlam komunikasi. Layanaan ini sering digunakan untuk berkomunikasi atau mengobrol di *internetWorld Wide Web*(WWW).
4. *Newsgroup* fasilitas ini digunakan untuk berkoferensi jarak jauh, sehingga pengguna dapat menyampaikan pendapat dan tanggapan dalaminternet.
5. *Download* adalah proses mengambil file dari komputer lain melalui internet ke komputerkita.
6. *Upload* adalah proses meletakkan file dari komputer kita ke komputer lain melalui internet.

7. *Transfer protocol* (FTP) fasilitas ini digunakan untuk melakukan pengambilan arsip atau file secara elektronik atau transfer file dari satu komputer ke komputer lain diinternet.
8. *Telnet* fasilitas ini digunakan untuk masuk ke sistem komputer tertentu dan bekerja pada sistem komputerlain.
9. *Ghoper* fasilitas ini digunakan untuk menempatkan informasi yang disimpan pada *internet server* dengan menggunakanhirarki.

Dari penjelasan diatas, *internet* merupakan kumpulan dari beberapa komputer yang terhubung dalam satu jaringan dan dapat diakses dari tempat yang sangat jauh.

2.1.7 *Website*

Website adalah sekumpulan link yang menyediakan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi yang berguna bagi pengguna. Sebuah website umumnya dibangun atas banyak halaman web yang saling berhubungan dan saling berkaitan. Hubungan antara satu halaman web dengan halaman web yang lainnya disebut dengan *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

Domain adalah nama unik yang dimiliki oleh sebuah institusi atau menjadi karakter dari sebuah perusahaan atau organisasi dalam memnggambarkan profil mereka sehingga bisa diakses melalui layanan media intemet, misalnya lintau.com, yahoo.com, google.com, ephi.web.id, dan lain-lain. Istilah lain yang

sering ditemui sehubungan dengan *website* adalah *homepage. Homepage* adalah halaman awal sebuah domain (Yuhefizar, Mooduto, & Hidayat, 2009).

2.2 Tinjauan Teori khusus

Tinjauan teori khusus memuat dan menjelaskan teori khusus yang berhubungan dengan objek pembahasan sesuai dengan judul penelitian skripsi.

2.2.1 *E-Comerce*

E-Commerce merupakan bagian dari *e-business*. Secara umum, seorang konsumen yang akan berbelanja online melalui internet memerlukan teknologi atau infrastruktur internet untuk mencari tahu tentang toko online atau webstore.(Kosasi, 2015)

E-Commerce adalah kegiatan-kegiatan bisnis yang menyangkut konsumen (*consumer*), manufaktur (*manufactures*), *services providers* dan padagang perantara (intermediaries) dengan menggunakan jaringan-jaringan *computer* (*computer networks*) yaitu internet. Secara umum, *e-commerce* meliputi aktifitas-aktifitas bisnis secara *online* untuk produk dan jasa yang bisa dibagi kedalam beberapa jenis *ecommerce*, yaitu:

a. Bussiness to Bussiness (B2B) Kelompok ini disebut sebagai transaksi antara perusahaan. Perusahaan, pemerintah, dan organisasi lainnya bergantung pada komunikasi antar komputer sebagai sarana bisnis yang cepat, ekonomis, dan dapat diandalkan, Perusahaan kecil saat ini sudah mulai tertarik dengan keuntungan yang diperoleh menggunakan model B2B ini. Transaksi pada B2B menggunakan EDI dan email untuk pembelian barang dan jasa, informasi dan

konsultasi. Selain itu juga digunakan untuk pengiriman dan permintaan proposal bisnis.

b. Bussiness to Customer (B2C) Kelompok ini disebut juga transaksi pasar. Pada transaksi pasar, konsumen mempelajari produk yang ditawarkan melalui publikasi elektronik, membelinya dengan electronic cash dan sistem secure payment, kemudian minta agar barang dikirimkan. Secara ringkas jenis *e-commerce* ini merupakan ecommerce yang melibatkan konsumen dengan merchant- nya secara langsung.

c. Customer to Bussiness (C2B) Untuk jenis yang satu ini seorang pelaku konsumen proyek dengan anggaran yang ditetapkan secara online dan dalam hitungan jam perusahaan meninjau persyaratan konsumen dan melakukan penawaran pada proyek tersebut. Konsumen dapat melakukan peninjauan tawaran dan memilih perusahaan mana yang akan menyelesaikan proyek mereka. ini bisa Anda lihat pelaku ini seperti di situs Freelancer.com

d. Customer to Customer (C2C) Kelompok ini disebut juga dengan marketplace, marketplace sebagai penyedia fasilitas untuk penjual dan pembeli melakukan transaksi (rekening bersama). Selain itu biasanya marketplace juga menyediakan layanan khusus untuk penjual mempromosikan barang atau produknya. *Marketplace* merupakan media *online* berbasis Internet *(web based)* tempat melakukan kegiatan bisnis dan transaksi antara pembeli dan penjual. Pembeli dapat mencari supplier sebanyak mungkin dengan kriteria yang diinginkan, sehingga memperoleh sesuai harga pasar. Sedangkan bagi

supplier atau penjual dapat mengetahui perusahaan-perusahaan yang membutuhkan produk atau jasa mereka.

2.2.2 *Event Organizer*

Sebagai sebuah *Event Organizer* tentu harus mampu untuk mengarahkan klientnya (pengguna jasa) agar acara yang direncanakan jelas arahnya. Pemaparan konsep kepada klien harus diajukan sejelas mungkin agar mereka mengerti apa yang di sampaikan. Dimana suatu konsep yang kreatif merupakan andalan setiap *Event Organizer* dalam mencari klien. Dengan adanya hal ini, bisnis dibidang *Event Organizer* tentunya sangat menjanjikan sehingga banyak sekelompok orang dalam sebuah organisasi maupun perusahaan memilih untuk membuka usaha di bidang *wedding organizer*.

Secara umum *Event Organizer* dapat diartikan sebagai sebuah badan organisasi penyedia layanan jasa yang bekerja untuk mengkordinasi sebuah acara. Kordinasi tersebut dimulai dari perencanaan acara, koordinasi ketika acara telah selesai diselenggarakan . Untuk bekerja di bidang seperti ini tentu saja haruslah orang-orang yang kreatif, komunikatif, ulet, dan pantang menyerah. (Sanjaya, 2016). Selain dari itu *Event Organizer* juga merupakan sekelompok orang, yang terdiri dari tim pelaksana, tim pekerja, tim produksi dan tim manajemen yang melaksanakan tugas operasional suatu program acara atau melakukan pengorganisasian untuk mewujudkan suatu program acara.

2.2.3 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah salah satu standar bahasa pemodelan yang banyak digunakan dan dipaki di dunia industri untuk menggambarkan atau mendefinisikan *requirement*, menggambarkan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *Unified Modeling Language* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *Unified Modeling Language* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak yang seddang berjalan atau yang akan dibangun.

Unified Modeling Language hanya berfungsi untuk melakukan pemodelan.terhadap system yang akan dibnagun Jadi penggunaan *Unified Modeling Language* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *Unified Modeling Language* paling banyak digunakan pada metodologi berorientasi objek (Rosa & Shalahuddin, 2013 : 133).

2.2.3.1 Sejarah *UML*

Unified Modeling Language dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan dan kemajuan yang aktif dari pemrograman berorientasi objek mulai berkembangt ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980 kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang

lainnya seperti *C objek*, *C++*,*Eiffel*, dan *CLOS*.Karena banyaknyametodologi-metodologi yang berkembang pesat saat itu, maka muncullah gagasan untuk membuat sebuah bahasa pemrogrman yang dapat dimengerti semua orang dan mudah dipahami dan diimplementasikan. Maka dibuat bahasa pemrogramanan yang merupakan kominasi dari beberapa konsep dan teori, seperti konsep *Object Modeling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML).*

Object Management Group (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *Unified Modeling Language* diakomodasi oleh *OMG* sehingga sampai saat ini *Unified Modeling Language* telah memberikan kontribusinya yang cukup besar dan berengaruh di dalam metodologi berorientasi objek dalam bahasa pemrograman dan hal-hal yang terkait di dalam dan yang memnbangunnya.

Secara fisik, *Unified Modeling Language* adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG *(Object Management Group)*. *Unified Modeling Language* terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, *Unified Modeling LanguageInfrastructure*, *Unified Modeling LanguageSuperstructure*, dan *Object Constraint Language* (OCL) (Rosa & Shalahuddin, 2013 : 136-140).

2.2.3.2 Diagram *UML*

Unified Modeling Language adalah bahasa pemodelan yang standar untuk lingkungan berorientasi obyek yang menggambarkan model system uang dibangun, yang berisi notasi notasi grafis yang *relative* sudah dibakukan (*open standard*). Sebagai sebuah sketsa, *Unified Modeling Language* berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dari sistem. *Unified Modeling Language* bisa juga berfungsi sebagai cetak biru karena sangat lengkap dan detil yang memudahkan pemahan pembaca. Dengan adanya cetak biru ini maka akan dengan mudah diketahui informasi detil tentang coding program (*forward engineering*) atau bahkan membaca program dan pengimplementasiannya kembali menurut diagram yang akan digambarkan*(reverse engineering*). Sebagai bahasa pemrograman, *Unified Modeling Language* dapat menterjemahkan diagram yang ada di *Unified Modeling Language* menjadi kode program yang siap untuk dijalankan (Lenti, 2014).

Unified Modeling Language terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategoritersebut.(Rosa & Shalahuddin, 2013 : 140),

1. *Structure diagram*, adalahisi atau kumpulan diagram yang dipakai untuk membangun suatu struktur statis dari sistem yang digambarkan. *Structure diagram* terdiri dari *class diagram, object diagram, component diagram, composite structure diagram, package diagram dan deploymentdiagram.*
2. *Behavior diagram* adalahisi diagram yang digunakan untuk menggambarkan kelakuan sistem atau kelakuan pengguna terhadap system yang digambarkan

sesuai alur yang tepat. Behavior itu sendiri terdiri dari diagram 3 diagram. *Use case diagram, Aktivity diagram, State Machine System.*

3. *Interaction diagram* adalahisi diagram yang diganbarkan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistempadasuatusistem.Diagram *Interactiondiagram*terdiridari*Sequence Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram.*

2.2.3.3 *Class* Diagram

Class Diagram adalah spesifikasi atau penjelassan yang akan menghasilkan objek dan merupakan inti dari pengembangan suatu sistem dan desain berorientasi objek. Class menggambarkan keadaan atau kondisi (atribut atau properti) suatu sistem, sekaligus menawarkan dan memberikan layanan untuk memanipulasi keadaan dan kondisi tersebut (metode atau fungsi) (Isa & Hartawan, 2017).

class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi(Rosa & Shalahuddin, 2013 : 141-147). Berikut penjelasan atribut dan *method* :

1. Atribut adalah variable-variabel yang dimiliki oleh suatukelas dari sistem.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki dari suatu sistem.

Berdasarkan usunan struktur kelas yang baik pada diagram kelas umumnya harus memiliki syarat dalam penamaan yang tepat agar dapat mudah dipahami bagi yang melihat kelas diagram:

1. Kelas main

   Kelas yang memiliki fungsi awal yan dijalankan pada saat awal.

2. Kelas tampilan sistem

   Mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas pendefinisian use case

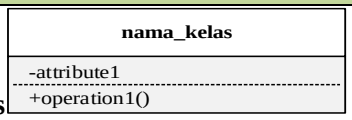
   Menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case.

4. Kelas pendefinisian data

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

   Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2. 1 Simbol- simbol yang ada pada diagram kelas

Simbol	Keterangan
Kelas 	Kelas yang ada pada struktursystem
Antarmuka 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
/association	Relasi antar kelas dengan *multiplicity*.
directed association 	Relasi antar kelas, asosiasi biasanya juga disertai dengan *multiplicity*.

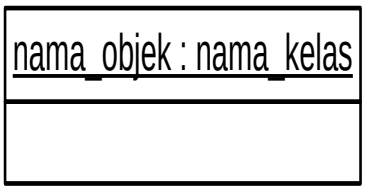
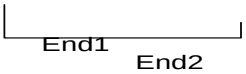
Dependency	Relasi kebergantungan antar kelas.
Aggregation	Relasi semua bagian (*whole-part*)

2.2.3.4 *Object* Diagram

Diagram objek juga berfungsi untuk mendefinisikan contoh nilai atau isi dari atribut tiap kelas Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. (Rosa & Shalahuddin, 2013 : 147).

Diagram ini untuk memperlihatkan satu prototipe atau kasus tertentu yang mungkin terjadi Diagram objek menunjukkan sekumpulan objek dan keterhubungannya. Diagram ini menunjukkan potongan statik dari instan-instan yang ada di diagram kelas. (Sopiah, 2012).

Tabel 2. 2 Simbol-simbol Yang ada pada diagram objek

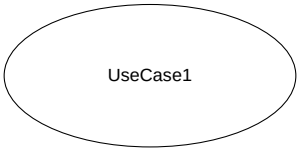
Simbol	Keterangan
Objek 	Objek dari kelas yang berjalan saat sistem dijalankan
Link 	Relasi antar objek

2.2.3.5 *UseCase* Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar. *Use Case* diagram merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat., Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua Hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case* (Rosa & Shalahuddin, 2013 : 155-158). *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

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

Tabel 2. 3 Simbol-simbol *use case*

Simbol	Keterangan
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*; biasanya dinyatakan dengan m enggunakan kata kerja di awal frase nama *use case*

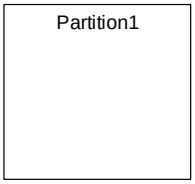
2.2.3.6 *Activity* Diagram

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sitsem. Yang perlu diperhatikan disini adalah bahwa Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa & Shalahuddin, 2013 : 161-162).

1. Rancangan prosessystem yang dibangun di mana setiap urutan aktivitas yang digambarkanmerupakan proses s sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan yang disajikan bagi pengguna system atau dari system yang dibangun.
3. Rancangan pengujian di mana setiap aktivitas atau kegiatan dari actor digambarkan dan dianggap memerlukan sebuah pengujian atau testing yang perlu didefinisikan kasus ujinya yang dimana tujuan dari pengujian dari rancangan ini adalah untuk mengetahui kelemahan dan kekurangan dari system yang dibangun untuk mendpat hasil yang signifikan yang dapat diambil keputusan untuk pengolahan data yang berguna bagi pengguna maupun untuk kebergunaan untuk tahap testing atau pengujian itu sendiri sehingga mempermudah kinerja wwaktu dan biaya yang dipakai sehingga perlunya tahapan testing dan pengujian yang baik dan sesuai dengan prosedur maupun aturan yang telah ada atau telah dibuat sebelumnya.

Berikut ini adalah Simbol-simbol yang ada pada *activity* diagram:

Tabel 2. 4 Simbol-simbol Activity Diagram

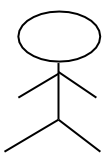
Simbbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
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.

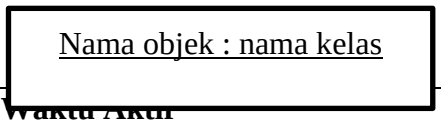
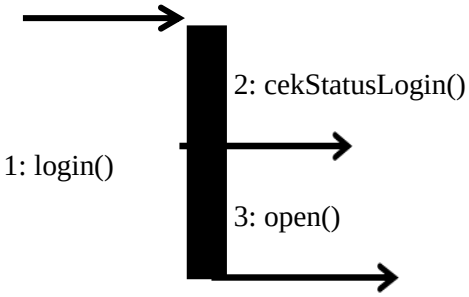
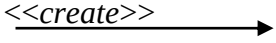
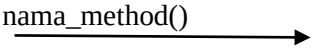
2.2.3.7 *Sequence* Diagram

Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banayak.

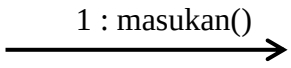
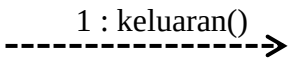
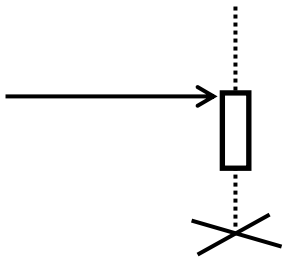
diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *massage* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu(Rosa & Shalahuddin, 2013 : 165-167),. Berikut adalah Simbol-simbol yang ada pada *sequence* diagram:

Tabel 2. 5 Simbol-simbol diagram sequence

Simbol	Keterangan
Aktor  Atau Aktor	Orang, proses, yang berinteraksi dengan sistem informasi yang akan dibuat simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.

Garis Hidup/ *lifetime*	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubungdengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya :  Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.
Pesan Tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe *call* 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelasobjek yang berinteraksi.

Tabel 2. 6 Lanjutan Simbol-simbol diagram sequence

Simbol	Keterangan
Pesan Tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panahmengarah 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 objek yang menerima kembalian.
Pesan Tipe *Destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada *create* maka ada *destroy*.

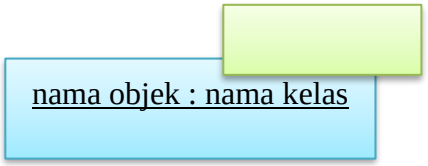
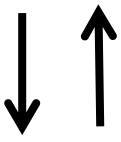
2.2.3.8 *Communication* Diagram

Diagram komunikasi mengelompokan *message* pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi/metode yang dijalankan antara objek yang satu dan objek lainnya secara keseluruhan, oleh karena itu dapat diambil dari jalannya interaksi pada semua diagram sekuenDiagram Komunikasi menggambarkan interaksi antarobjek/bagian dalam bentuk urutan pengiriman pesan.. Penomoran metode

dapat dilakukan berdasarkan urutan dijalankannya (Rosa & Shalahuddin, 2013 : 168-169).

Berikut simbol-simbol yang ada pada *communication* diagram:

Tabel 2. 7 Simbol-simbol pada communication diagram

Simbol	Keterangan
Objek 	Objek yang melakukan interaksi pesan,
Link 	Relasi antar objek yang menghubungkan objek satu dengan yang lainnya atau dengan dirinya sendiri. 
Arah Pesan / *stimulus* 	Arah pesan yang terjadi, jika pada suatu *link* ada dua arah pesan yang berbeda maka arah juga digambarkan dua arah pada dua sisi *link* 

2.2.4 Flowchart

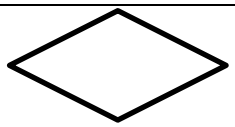
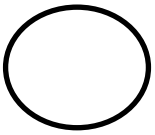
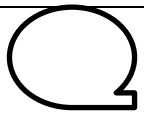
Flowchart atau bagan alir adalah suatu bagan yang berisi simbol-simbol grafis yang menunjukkan arah aliran kegiatan dan data-data yang dimiliki program sebagai suatu proses eksekusi. *Flowchart* pada umumnya menggambarkan suatu kegiatan dari interaksi yang ada dalam sistem, melalui proses (*input*) yaitu kegiatan atau masukan yang terjadi diawal sistem kemudian menggambarkan tahap proses yaitu kegiatan yang berlangsung setelah adanya proses (*input)* yang terjadi kepada sistem kemudian adanya pengolah yang digambarkan melalui bagan *flowchart* dengan simbol-simbol yang telah ada pada aturan *flowchart diagrams,* kemudian proses *output* yaitu keluaran yang terjadi setelah proses *Input* dan *process* yang secara keseluruhan diagram *flowchart* merepresentasiakan seluruh kegiatan atau aktivitas yang terjadi dalam sistem melalui bagan-bagan tertentu yang dapat dilihat secara visual. Simbol-simbol dalam *flowchart* itu sendiri mempunyai arti yang berbeda setiap simbol sehingga dalam pemakain atau penggunaan simbol *flowchart* harus menyesuaikan dengan nama kegiatan atau aktivitas yang akan dilakukan atau yang akan terjadi dalam sistem (Rasim, Setiawan, & Rahman, 2008).

Bagian alir program (*Flowchart*) adalah bagian yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir".(Drs. Katen Lumbanbatu & Novriyeni, S.Kom., 2013).

Bagan alir program (Program *Flowchart*) merupakan bagan yang menjelaskan secara rinci, langkah-langkah dari proses program. Bagan alir program disebut dari derivikasi bagan alir sistem (Toibah Umi Kalsum1, 2012).

Bentuk-bentuk yang digunakan dalam pembuatan flowchart mempunyai arti-arti khusus, yaitu sebagai berikut:

Tabel 2. 8 Simbol Flowchart

Simbol	Keterangan
	Memulai atau Mengakhiri Program
	Proses (Menulis atau Menjalankan) Program
	Masukan atau Keluaran Pengambilan
	Pengambilan Keputusan atau Pengujian Program
	Penghubung Program Magnetic
	Magnetic Tape Magnetic
	Magnetic Disk Arah
	Arah aliran
	Tampilan/Penyajian Hasil Pemprosesan Data

2.2.5 DFD (*Data Flow Diagram*)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.(Afyenni, 2014)

Data Flow Diagram (DFD) adalah diagram yang menggunakan atau menggunakan lambang notasi-notasi untuk menggambarkan suatu sistem yang telah ada atau sistem baru atau system yang akan dibangun dan yang akan dikembangkan secara logika, tanpa mempertimbangkan atau memperhitungkan lingkungan fisik dimana data tersebut mengalir dan berjalan atau lingkungan fisik dimana data tersebut akan disimpan.(Giyanmandiri, 2016)

Ada empat buah simbol pada Data Flow Diagram, yang masing-masingnya digunakan untuk mewakili (Afyenni, 2014) :

a. Data *flow* (arus data), digunakan untuk menunjukkan arus dari data yang dapat berupa: masukan untuk sistem ataupun hasil dari proses sistem. Arus data sebaiknya diberi nama yang jelas dan mempunyai arti. Didalam menggambarkan arus data di DFD perlu diperhatikan beberapa konsep berikut:
    1. Konsep paket dari data (*packet of data*). Bila dua atau lebih data mengalir dari suatu sumber yang sama ketujuan yang sama, maka dianggap sebagai suatu arus data tunggal.
    2. Konsep arus data menyebar. Menunjukkan sejumlah tembusan dari arus data yang sama dari sumber yang sama ketujuan berbeda.

4. Konsep arus data mengumpul *(converging data flow*) Menunjukkan beberapa arus data yang berbeda bergabung bersama- sama menuju ke tujuan yang sama.
5. Konsep sumber dan tujuan arus data Semua arus data harus dihasilkan dari suatu proses atau menuju ke suatu proses.

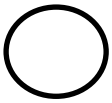
b. External entity (kesatuan luar) atau *boundary* (batas sistem), digunakan untuk menyatakan: suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang dikembangkan; orang atau sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan; suatu organisasi atau orang yang berada di luar organisasi misal: langganan, pemasok; sistem informasi yang lain di luar sistem yang sedang dikembangkan; sumber asli suatu transaksi; penerima akhir dari suatu laporan yang dihasilkan oleh sistem.
c. Proses (*process*), digunakan untuk menunjukkan kegiatan atau kerja yag dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang keluar dari proses. Suatu proses harus menerima arus data dan menghasilkan arus data.
d. Data *store* (simpanan data), digunakan untuk menunjukkan simpanan dari data yang dapat berupa: suatu file atau database di sistem komputer; suatu arsip atau catatan manual; suatu kotak tempat data di meja seseorang; suatu tabel acuan manual; suatu agenda atau buku.

Data fl merupakaow diagram merupakan model dari sistem untuk menggambarkan dan menjelaskan aliran sistem ke modul yang lebih kecil sehingga mempermudah dalam mengartikan dan menjelaskan tentang aliran

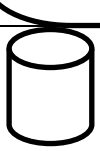
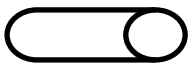
system informasi yang dibangun. Salah satu keuntungan menggunakan diagram data flow diagram adalah memudahkan engguna atau *user* yang kurang menguasai bidang ilmu komputer untuk mengerti sistem yang akan dikerjakan dan yang akan dibangun.

Berikut ini merupakan gambar *data flow diagram* yang umum digunakan dalam perancangan sistem:

Tabel 2. 9 Simbol DFD 1

Simbol	Nama	Keterangan
	Terminal	Awal atau akhir
	Input/Output	Mempresentasikan input data atau output data yang diproses atau informasi
	Penghubung	Keluar atau masuk ke bagian yang lain dalam halaman yang sama
	Anak panah	Mempresentasikan alur kerja
	Penjelasan	Dugunakan untuk komentar tambahan
	Proses	Mempresentasikan operasi
	Keputusan	Keputusan dalam program
	Predefined proses	Rincihan operasi yang berada di tempat lain
	Preparation	Pemberian harga awal

Tabel 2. 10 Symbol DFD 2 Lanjutan

Simbol	Nama	Keterangan
	Dokumen	I/O yang dicetak
	Manual Operaion	Operasi manual
	Online Storage	I/O penyimpan akses lngsung
	Communication link	chanel koneksi
	Punched Tape	I/O penghubung
	Punched Card	I/O kartu penghubung
	Magnetic Tape	I/O pita magnetic tape
	Magnetic Disk	I/O magnetic disk
	Magnetic Drum	I/O Magnetic Drum

2.2.6 HTML (*Hyper Text Markup Language*)

Sejarah html berawal pada tahun 1980 ketika IBM berniat untuk membuatkan suatu bahasa kode untuk menggabungkan teks dengan pemformatan

agar mengenali elemen dokumen. Bahasa yang menggunakan tanda-tanda ini dinamakan *Markup Language*. Namin pihak IBM memberi nama *Generalized Markup Language* (GML).(*Hyper Text Markup Language*) bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman website. HTML digunakan untuk manampilkan berbagai informasi didalam sebuah penjelajah *web* internet dan *formatting hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi (Saputra, 2012). Dokumen html memiliki struktur yang harus kita ikuti aturan pembuatannya. Beberapa elemen-elemen htm yang wajib ada pada html adalah berikut:

1. Elemen HTML

   Elemen HTML merupakan tag dasar apabila kita ingin memulai suatu dokumen html. Secara logika, jika kita menemukan tag ini, berarti secara jelas dapat didefenisikan bahwa dokumen ini merupakan perintah suatu dokumen html.

2. Elemen *Head*

   Head merupakan tag berikutnya setelah elemen html, yang berfungsi untuk menuliskan keterangan tentang dokumen web yang akan ditampilkan. Elemen ini nantinya akan diakhiri dengan tanda penutup <head>.

3. Elemen *Title*

   Elemen *title* merupakan suatu elemen yang harus dituliskan didalam elemen *head* yang digunakan untuk memberikan judul/ informasi pada *caption*

browser web tentang topik/ tema atau judul dari suatu dokumen web yang ditampilkan pada browser.

4. Elemen *Body*

   Merupakan bagian utama dalam dokumen web. Jika kita ingin menampilkan suatu teks atau informasi atau yang dikenal dengan sebutan konten, maka kita harus meletakkan teks tersebut pada elemen *body*.

2.2.7 CSS *(Cascading Style Sheet)*

Cascading Style Sheet merupakan singkatan dari *Cascading Style Sheet*. *CSS* biasa digunakan dalam dokumen *HTML* untuk menciptakan suatu *style* yang dipakai untuk mengatur penampilan elemen *HTML*. Dengan menggunakan *style*, suatu elemen dapat diformat dengan fitur yang jauh lebih kaya daripada yang disediakan oleh elemen *HTML* itu sendiri. Sebagai contoh, pengaturan seperti warna tulisan bisa ditangani melalui *style* tanpa melibatkan tag *HTML* yang berfungsi untuk mengatur warna (Maudi, Nugraha, & Sasmito, 2014).*Cascading Style Sheet*atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatubahasapemrograman*web*yangdigunakan untuk memperindah tampilan halaman *website*(situs), mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapi, terstruktur, dan seragam". (Sagita & Sugiarto, 2016).

*Cascading Style Sheet*dapat mengendalikan ukuran gambar, warna bagian tubuh teks, warna tabel, ukuran border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameterlainnya (Muarie, 2015).

2.2.8 PHP *(Personal Home Page)*

Personal Home Page adalah bahasa pemrograman skrip sederhana yang digunakan untuk pemrosesan HTML *Form* di dalam halaman *web*. Strukturnya sangat sederhana sehingga *Personal Home Page* dapat dengan mudah dipelajari programmer pemula bahkan orang tanpa latar belakang Teknologi Informasi. Hal inilah yang menyebabkan *Personal Home Page* sangat cepat populer di kalangan pengembang aplikasi *web*. Membuat program menggunakan PHP itu mudah, cukup sediakan saja sebuah program editor teks sederhana untuk menuliskan programnya, seperti *Notepad* (*Windows*) dan *vieditor* (*Linux*), atau program editor yang lebih *advance*, seperti *EditPlus*, *Notepad++*, atau *Dreamweaver*(Hastanti et al., 2015).*Personal Home Page* adalah pemrograman *interpreter* yaitu prosespenerjemahanbariskodesumbermenjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. *Personal Home Page* merupakan singkatan dari *Hypertext Preprocessor*, dan merupakan bahasayang disertakan dalam dokumen HTML sekaligus bekerja di sisi *server* (*server-side HTML embedded scripting*) (Sagita & Sugiarto, 2016).

Beberapa kelebihan dari *Personal Home Page* itu sendiri antara lain:

a. Mudah dipelajari.
b. Mampu lintas*platform*.
c. *Free* (gratis) bersifat *opensource*
d. PHP memiliki tingkat akses yangcepat
e. Didukung oleh beberapa macam*web* server.
f. Mendukungdatabase.

2.2.9 XAMPP

XAMPP menyediakan antar muka *control panel* tersendiri yang dapat digunakan untuk menjalankan semua *service* (paket *software* pendukung) yang telah terinstal. Pada sistem operasi windows, *controlpanel* dapat diakses melalui menu [Start]→ [Program] →[Apachefriends] → [xampp] →[control xampp server panel]. XAMPP merupakan paket PHP berbasis *open source*. Informasinya dapat diperoleh di website resminya: http://www.apachefriends.com. XAMPP membantu memudahkan dalam mengembangkan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket *software* berbeda kedalam satu paket. Adapun lisensi masing-masing paket *software* tersebut dapat ditemukan didirektori \xampp\licence.

Pada web server (lokal komputer, tidak di server internet sesungguhnya) pada XAMPP, akan menyediakan satu *folder* kerja yang bernama htdocs. Pada paket ini, *folder* kerja tersebut dapat ditemukan pada *subfolder* C:\..\XAMPP (sesuai lokasi dimana menyimpan hasil instalasinya) (Maudi et al., 2014).

2.2.10 MySQL (*My Structure Query Language*)

Pada saat ini MySQL merupakan basis data server yang sangat terkenal di dunia, semua itu karena bahasa dasar yang digunakan untuk mengakses basis data yaitu SQL (*Structure Query Language*). Dengan menggunakan SQL, proses pengaksesan basis data lebih *user-friendly* dibandingkan dengan yang lain, misalnya *dBAse* atau *clipper* karena mereka masih menggunakan perintah-perintah pemrograman murni (Maudi et al., 2014).

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL yang *multithread*, multi-*user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah implementasi dari manajemen basis data relasional (RDBMS).

2.2.11 *JavaScript*

JavaScript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, bahasa ini adalah bahasa skrip pertama untuk web. Bahasa pemrograman ini memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi client, yang artinya di sisi browser bukan di sisi server web. JavaScript mampu mengakses Document Object Model serta melakukan manipulasi terhadapnya (Yenni & Shamir, 2012). *JavaScript* adalah bahasa skrip yang ditempatkan pad kode HTML dan diproses pada sisi klien. Dengan adanya bahasa ini maka kemampuan dokumen HTML menjadi lebih luas. Sebagai contoh, digunakan untuk validasi masukan pada formulir sebelum diproses ke tahap selanjutnya. Bisa untuk membuat permainan interaktif dan juga bisa untuk menambah desain*web* (Muarie, 2015).

.

2.2.12 *Extreme Programing*

Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas

maupun terjadi perubahan – perubahan requirement yang sangat cepat(Prabowo & Artwodini, 2013: 477). Metode *Extreme Programming* sering juga dikenal dengan metode XP.Metode ini dicetuskan oleh Kent Beck, seorang pakar software engineering. Extreme programming adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel.(Fatoni & Dwi, 2016)

Nilai dasar metode extreme programming :

1) *Communication* : Memfokuskan komunikasi yang baik antara programmer dengan *user* maupun antar *programmer*.

2) *Courage* : Pengembang perangkat lunak harus selalu memiliki keyakinan, keberanian dan integritas dalam melakukan tugasnya.

3) *Simplicity* : Lakukan semua dengan sederhana.

4) *Feedback* : Mengandalkan feedback sehingga dibutuhkan anggota tim yang berkualitas.

5) *Quality Work* : Proses berkualitas berimplikasi pada perangkat lunak yang berkualitas sebagai hasil akhirnya.

Untuk mencapai kesederhanaan, XP membatasi pengembang perangkat lunak melakukan perancangan hanya untuk kebutuhan- kebutuhan yang sifatnya mendesak alih-alih melakukan perancangan kebutuhan-kebutuhan yang diperlukan di masa depan. Tujuannya adalah untuk menciptakan rancangan yang sederhana yang dapat dengan mudah diimplementasikan dalam bentuk kode-kode program secara cepat. jika rancangan tersebut selanjutnya harus ditingkatkan,

rancangan yang bersangkutan dapat di-refaktorisasi di waktu yang lain (Korowotjeng, Sengkey, Paturusi, Tuturoong, & Kom, 2014: 4).

Pemrograman Ekstreme menggunakan suatu pendekatan 'berorientasi objek' sebagai paradigma pengembangan yang diinginkan dan mencakup di dalamnya seperangkat aturan dan praktik-praktik yang terjadi dalam konteks empat kegiatan kerangka kerja: perencanaan, perancangan, pengkodean, dan pengujian.

Perencanaan Kegiatan

Perencanaan biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan untuk mengumpulkan kebutuhan-kebutuhan yang memungkinkan anggota teknis tim XP memahami konteks bisnis untuk perangkat lunak yang akan dikembangkan dan untuk merasakan perlunya

output, fitur-fitur utama, dan fungsionalitas.Aktivitas- aktivitas mendengarkan pada dasarnya mengarah kepada pembuatan serangkaian "cerita" (juga disebut user stories) yang menggambarkan keluaran yang diperlukan, fitur-fitur, dan fungsionalitas-fungsionalitas yang akan dibangun menggunakan perangkat lunak yang akan dikembangkan. Setiap cerita umumnya ditulis oleh para pelanggan dan diletakkan pada kartu indeks.pelanggan- pelanggan memberikan suatu nilai (yaitu, suatu prioritas) pada cerita-cerita tertentu berdasarkan seluruh nilai bisnis dari fitur.Fungsi Anggota tim XP kemudian menilai setiap cerita dan menetapkan biayanya yang diukur dalam bentuk minggu- minggu yang diperlukan untuk melakukan pengembangan cerita tersebut. jika cerita ini diperkirakan membutuhkan lebih dari tiga minggu pengembangan, pelanggan akan diminta untuk membagi cerita tersebut ke dalam cerita-cerita kecil dan perhitungan-

perhitungan nilai dan biaya terjadi lagi. Penting untuk diperhatikan bahwa cerita baru dapat ditulis setiap saat.Pelanggan dan pengembang kemudian bekerja sama untuk memutuskan bagaimana mengelompokkan cerita kedalam kelanjutannya untuk dikembangkan oleh tim XP. Ketika komitmen dasar (kecocokan atas cerita yang dimasukkan, tanggal pengiriman, dan hal-hal lainnya yang terkait dengan proyek perangkat lunak) ini dirilis,tim XP mengatur cerita yang akan dikembangkan dalam sate dari tiga cara : (1) semua cerita akan segera dilaksanakan, (2) cerita yang memiliki nilai bisnis tertinggi akan dipindahkan ke dalam jadwal dan dilaksanakan pertama kali, atau (3) cerita yang paling berisiko akan dijadwalkan dan akan dilaksanakan pertama kali.

Perancangan, Perancangan XP dengan ketat mengikuti prinsip "tetap sederhana". Sebuah hasil perancangan yang sederhana selalu lebih disukai daripada gambarangambaran yang lebih kompleks. Selain itu, perancangan XP akanmemberikan panduan implementasi untuk suatu cerita ketika cerita itu ditulis, tidak kurang, tidak lebih. Rancangan-rancangan dan fungsionalitas-fungsionalitas tambahan (karena pengembang menganggap nantinya akan diperlukan) tidak terlalu disarankan.

Extreme Programing mendorong penggunaan kartu CRC sebagai mekanisme yang efektif untuk berpikir tentang perangkat lunak dalam konteks berorientasi objek.Kartu CRC (*class-responsibility- collaborator*) digunakan untuk mengidentifikasi dan mengatur kelas-kelas dalam konteks “pemrograman berorientasi objek" yang relevan dengan peningkatan perangkat lunak saat ini. Tim XP selanjutnya akan melakukan latihan perancangan. Kartu CRC adalah

satu-satunya produk kerja perancangan yang dihasilkan sebagai bagian dari proses pengembangan perangkat lunak cepat: XP. Jika masalah perancangan yang sulit ditemui sebagai bagian dari perancangan suatu cerita, metode pengembangan cepat PX menyarankan pembuatan langsung dari prototipe operasional dari bagian perancangan tersebut.Disebut sebagai solusi spike, prototipe perancangan diimplementasikan dan selanjutnya dievaluasi. Tujuannya adalah untuk mengurangi risiko-risiko yang akan timbul ketika pelaksanaan yang sesungguhnya dimulai dan untuk memvalidasi keaslian perkiraan untuk cerita yang berisi masalah-masalah perancangan.

Refaktorisasi pada dasarnya adalah proses mengubah sistem perangkat lunak sedemikian rupa sehingga tidak mengubah perilaku eksternal kode, namun memperbaiki struktur internal yang ada di dalamnya. Ini merupakan cara yang disiplin untuk membersihkan kode dan memodifikasi/menyederhanakan rancangan internal, yang pada gilirannya akan meminimalkan kemungkinan akan munculnya kesalahan-kesalahan program. Pada intinya, ketika Anda melakukan refactor atau refaktorisasi, Anda pada prinsipnya memperbaiki rancangan kode-kode program Setelah kode-kode program itu ditulis. Karena perancangan perangkat lunak pada rekayasa perangkat lunak cepat (XP) hampir-hampir tidak menggunakan notasi dan, kalaupun ada, hanya menghasilkan beberapa saja, tidak menghasilkan produk-produk kerja selain kartu-kartu CRC dan solusi-solusi spike, maka perancangan pada XP dipandang sebagai buatan sementara yang dapat dan harus terns menerus diubah wiring majunya pelaksanaan konstruksi perangkat lunak. Tujuan dari refaktorisasi pada dasarnya adalah untuk

mengendalikan modifikasi-modifikasi yang perlu dilakukan dengan cara menyarankan perubahan perancangan kecil yang "secara radikal dapat meningkatkan kualitas rancangan".

Pengkodean.

Setelah cerita dikembangkan dan karya rancangan awal dilakukan, tim perangkat lunak cepat (XP) tidak langsung beralih ke kode-kode program, tetapi lebih dulu akan mengembangkan serangkaian unit pengujian yang akan menjalankan setiap cerita yang akan disertakan pada rilis yang ada (peningkatan perangkat lunak)."

Setelah unit pengujian dibuat, pengembang perangkat lunakakan lebih mampu berkonsentrasi pada apa yang harus diimplementasikan supaya lulus dari unit pengujian tersebut. Tidak ada yang ditambahkan (tetap sederhana).Ketika kode-kode program telah selesai dituliskan, kode-kode program tersebut dapat langsung diuji menggunakan unit pengujian yang telah dirancang sebelumnya, sehingga bisa langsung memberikan umpan balik kepada para pengembang.

Konsep kunci selama kegiatan pengkodean adalah pemrograman berpasangan (*pair programming*). XP menyarankan bahwa dua orang pemrogram seharusnya bekerja sama pada satu komputer *workstation* untuk menuliskan kode-kode program untuk suatu cerita. Hal ini menyajikan mekanisme untuk pemecahan masalah yang dapat dilakukan secara real-time dan jaminan kualitas real-time (kode-kode program langsung ditinjau ketika ia dibuat). Hal ini juga membuat para pengembang perangkat lunak dapat berfokus pada masalah yang dihadapi.Dalam praktiknya, setiap orang mengambil peran yang sedikit berbeda.

Misalnya, satu orang mungkin berpikir tentang rincian pengkodean dari bagian tertentu dari suatu perancangan, sementara orang yang lain memastikan standar pengkodean (merupakan bagian yang diharuskan dari XP) diikuti atau kode untuk cerita memenuhi unit pengujian yang telah dikembangkan untuk melakukan validasi kode terhadap cerita.

Pengujianbahwa pembuatan unit pengujian sebelum pengkodean dimulai merupakan elemen kunci dari pendekatan pengembangan perangkat lunak cepat XP. Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi (sehingga mereka dapat dijalankan dengan mudah dan dapat dijalankan berulang kali).Hal ini mendorong strategi pengujian regresi terhadap kode- kode program setiap kali kode-kode program tersebut diubah (yang seringkali dikaitkan dengan falsafah refactoring XP).Sebagai unit pengujian individu yang diatur dalam "rangkaian pengujian universal", pengujian integrasi dan validasi atas sistem/perangkat lunak pada umumnya dapat terjadi. Hal ini menyajikan bagi tim perangkat lunak XP suatu indikasi kemajuan dan juga dapat memberikan peringatan awal jika terjadi sesuatu yang keliru. Wells menyatakan: "Memperbaiki masalah-masalah kecil setiap beberapa jam membutuhkan waktu lebih sedikit daripada memperbaiki masalah besar sebelum batas waktunya."

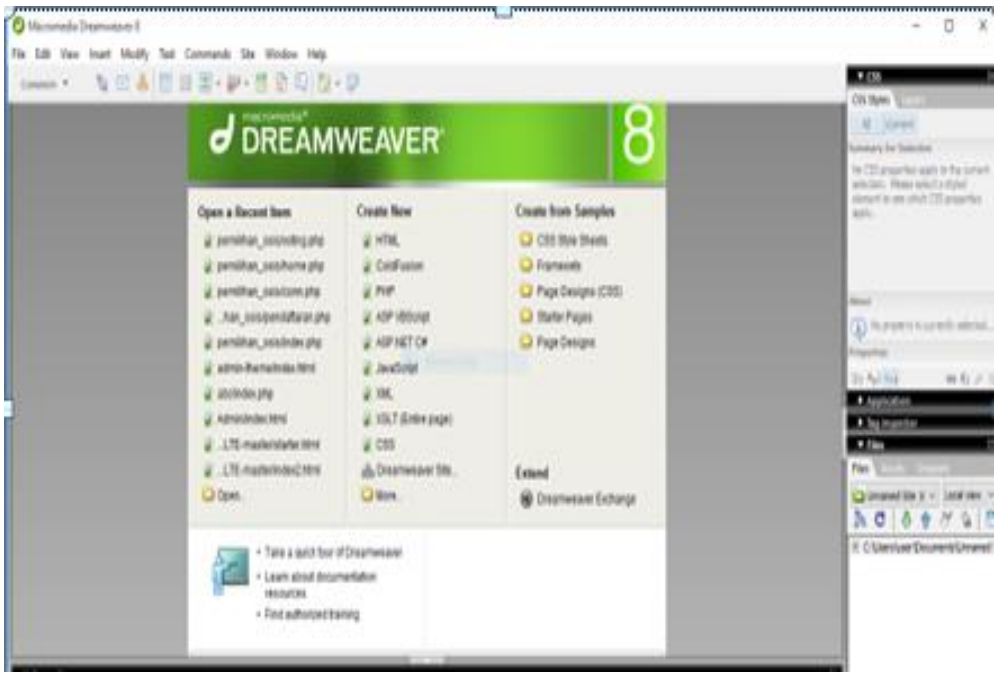
Uji kelayakan XP, sering juga disebut uji pelanggan, dirincioleh para pelanggan dan pada dasarnya berfokus pada fitur- fitur dan fungsionalitas-fungsionalitas sistem/perangkat lunak secara keseluruhan yang dapat terlihat dan

ditinjau kembali oleh para pelanggan.Uji kelayakan berasal dari cerita pengguna yang telah diimplementasikan sebagai bagian dari suatu rilis perangkat lunak.

2.2.13Dreamweaver

Menurut jurnal (Destiningrum & Adrian, 2017), Adobe Dreamweaver adalah : "aplikasi desain dan pengembangan web yang menyediakan editor WYSIWYG visual (bahasa sehari-hari yang disebut sebagai *Design view*) dan kode editor dengan fitur standar seperti *syntax highlighting, code completio*n, dan *code collapsing* serta fitur lebih canggih seperti *real-time syntax checking* dan *code introspection* untuk menghasilkan petunjuk kode untuk membantu pengguna dalam menulis kode".Adobe Dreamweaver merupakan program penyunting halaman web keluaran Adobe Systems yang dulu dikenal sebagai Macromedia Dreamweaver keluaran Macromedia. Program ini banyak digunakan oleh pengembang *web* karena fitur-fiturnya yang menarik dan kemudahan penggunaannya.

Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh *AdobeSystems* yaitu versi 8. Versi terakhir Dreamweaver keluaran *AdobeSystems* adalah versi 10 yang ada dalam Adobe Creative Suite 4 (sering disingkat Adobe CS4) (Maudi et al., 2014).



Gambar 2. 3 Tampilan Macromedia Dreamweaver

2.3 Penelitian terdahulu

Penelitian terdahulu ini menjadi salah satu acuan dan bagian terpenting bagi penulis dalam melakukan pembangunan system yang yang akan dibangun dan supaya penelitian ini dapat berjalan denganbaik sehinnga perlunya bahan referensi tanbahan sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan peneletian yang berkaitan erat dengan penelitian yang dilakukan penulis sendiri dan sesuai dengan judul. Namun penulis mengangkat beberapa penelitian sebagai referensi dan sebagai bahan teori dalam memperkaya bahan kajian pada penelitian penulis. Berikut ini merupakan penelitian terdahulu berupa beberapa jurnal terkait yang mememiliki kaitan erat dengan penelitian yang dilakukan penulis itu sendiri.

Untuk memudahkan pemahan penulis berikut ini adalah penelitian terdahulu yang didapat dari beberapa jurnal sebagi bahan referensi yang dilakukan penulis untuk mendukung teori yang dipakai penulis untuk membangun system yang akan dibuat.

Tabel 2. 11 Tabel Penelitian Terdahulu

No	Tahun	Peneliti	Judul	Metode	Kesimpulan	Penerbit
1	2015	, Ernes Cahyo Nugroho , Didit Gunawan	Sistem Informasi Sewa Berbasis Web Di Surakarta	Metode *Waterfall*	Membantu memberikan informasi kepada pengguna system tentang layanan yang tersedia fitur yang ditawarkan, sehingga memepermudah pengguna dalam melakukan transaksi dan lebih memnghemat biaya yang dikeluarkann pengguna itu .	STMIK Surakarta

Tabel 2. 12 Lanjutan Tabel Penelitian Terdahulu

No	Tahun	Peneliti	Judul	Metode	Kesimpulan	Penerbit
2	2013	Hamzah, Joko Triyono Guridno Adi Saputro, Amir	Sistem Informasi Pengolahan Data s Mengguna kan *Framework Codeignite r* Di Kelurahan Demangan Kecamatan Gondokusu man Yogyakarta	*Model View Controller*	System yang dibangun memiliki fitur pencarian yang memudahkan konsumen	Teknik Informatika Institut Sains Yogyakarta
3	2016/ 2442-4943	Feby Oktarista Andriawa n	Aplikasi Sistem Informasi Pencarian Tempat Di Kota Bandung Berbasis *Android*	Metode *Waterfall*	Pencarian informasi yang disediakan lebih efektif dan memmpermmuda h pengguna dalam meneukan lokasi dengan bantuan web yang dibuat, denagan adanay media website berpengaruh besar bagi pemilik dalam mempromosikan	STMIK Mardira Bandung

Tabel 2. 13 Lanjutan Tabel Penelitian Terdahulu

No	Tahun	Peneliti	Judul	Metode	Kesimpulan	Penerbit
4	2014	George Mwenda Njiru, Arphaxad Nguka Owange Henry Peter Gommans	*Rental Housing Management System*	*waterfall model*	*the software can be used as an inventory system to provide a frame work that enables the mangers to make reasonable transactions made within a limited time frame.*	*Supply Chain Professional/Lecturer), Tradewinds Kenya Limited Scholar*