

BAB II

TINJAUAN PUSTAKA

2.1 E-Commerce

Transaksi dalam dunia internet diartikan sebagai *e-commerce*. Terdapat berbagai macam jenis transaksi yang bisa diselesaikan dengan media internet, seperti: transaksi berupa blog, transaksi di forum, media sosial, toko *online* dan *e-marketplace*. Karakteristik yang dimiliki *e-commerce* yaitu: internet merupakan media perantara untuk mengintegrasikan proses bisnis baik penjualan, pembelian, pesediaan dan pemesanan. Tanpa internet sebagai media perantara, proses bisnis hanya berjalan secara manual dan biaya yang digunakan lebih besar [8].

E-commerce menurut M. Suyanto, “*E-commerce* adalah sebuah konsep baru yang dapat diartikan sebagai proses jual beli barang atau jasa di Internet, atau proses jual beli produk, jasa dan informasi melalui jaringan informasi termasuk Internet.”. Beberapa orang melihat istilah perdagangan (*commerce*) sebagai transaksi antar perusahaan mitra. Oleh karena itu, istilah *electronic commerce* tampaknya sempit bagi banyak orang. Oleh karena itu, banyak yang lebih suka menggunakan istilah *e-business*, yang mengacu pada definisi *e-commerce* yang lebih luas daripada sekedar jual beli.

1. Manfaat E-Commerce

a. Manfaat bagi perusahaan

- Memperluas jangkauan penjualan.
- Mengurangi biaya operasional.

b. Manfaat bagi konsumen

- Mempermudah proses pembelian dan pembayaran karena dapat dilakukan secara online.
- Lebih simpel memilih berbagai banyak macam produk meski tidak datang langsung ke toko.

c. Manfaat bagi masyarakat

- Memungkinkan orang untuk bekerja di dalam rumah serta tidak harus keluar rumah untuk berbelanja.
- Memungkinkan sejumlah barang di jual menggunakan harga lebih murah sehingga orang yang kurang mampu bisa membeli lebih banyak yang pada gilirannya akan meningkatkan taraf hidup mereka.

2. Jenis-jenis E-commerce

- a. *Business to Business* (B2B). jenis *e-commerce* ini mencakup transaksi antar organisasi yang dilakukan di pasar elektronik.
- b. *Business to Costumer* (B2C) adalah transaksi ritel dengan pembeli individu.
- c. *Customer to Customer* (C2C). Konsumen menjual produk eksklusif ke konsumen lain atau mempromosikan layanan langsung pada Internet.
- d. *Customer to Business* (C2B). Individu yang menjual produk / jasa kepada organisasi, individu yang mencari penjual untuk berinteraksi dan menyelesaikan transaksi.

3. Keuntungan *Electronic Commerce* (*E-Commerce*). Terdapat beberapa keuntungan yang bisa diambil dari penggunaan *E-commerce* yaitu :

1. Pelanggan *E-Commerce* dapat berbelanja sepanjang hari terserah pelanggan melakukan transaksi dimanapun berada.
2. *E-commerce* memberi pelanggan lebih banyak pilihan, mereka mampu memilih aneka macam produk berasal banyak vendor. Dengan mengunjungi banyak wilayah dan membuat perbandingan cepat, *e-commerce* menyediakan produk dan layanan murah kepada pelanggan[9].

2.1.1 E-marketplace

E-marketplace merupakan sistem informasi antar organisasi yang mempertemukan pembeli serta penjual dalam jumlah besar sehingga memperluas pilihan bagi pembeli dan penjual, dan menyampaikan fakta mengenai harga dan produk serta mampu merampungkan transaksi melalui saluran komunikasi elektronik

dan bertujuan untuk mengurangi biaya transaksi antar pihak yang bersangkutan secara bersamaan. *E-marketplace* sebagai pihak ketiga yang memediasi interaksi antara penawaran dan permintaan, mendorong interaksi antara pembeli dan penjual untuk menegosiasikan harga, meningkatkan kepercayaan dan membantu dalam proses transaksi.

E-marketplace ada pada aneka macam bentuk serta bisa mendukung banyak kebutuhan usaha yang tidak sinkron. *E-marketplace* dapat melayani pasar vertikal maupun pasar horisontal, dapat dimiliki oleh pembeli, penjual atau bahkan pihak independen, dapat memungkinkan menjadi pasar dengan harga tetap, fleksibel atau kombinasi keduanya, pasar dapat terbuka untuk umum atau hanya terbuka untuk sebagian golongan. Model *e-marketplace* dapat dikategorikan menjadi empat kategori, diantaranya sebagai berikut :

1. *Consortia*

Kelompok perusahaan dalam suatu industri menjalankan dan memiliki *e-marketplace* dan dapat membeli serta menjual satu sama lain antar perusahaan.

2. *Private*

Satu organisasi memiliki dan menjalankan *e-marketplace* sendiri. Hal ini bisa menjadi menjual sisi *e-marketplace*, dimana organisasi menjual produk kepada pembeli yang memenuhi syarat atau sisi pembeli dimana pembeli dapat berpartisipasi sebagai pemasok.

3. *Public / Independent*

Suatu organisasi independen dapat memiliki dan menjalankan *e-marketplace* sehingga pihak lain tidak memiliki kepemilikan terhadap *e-marketplace*.

4. *Community*

Kelompok masyarakat atau organisasi pemerintah yang memiliki kepemilikan dan menjalankan *e-marketplace* untuk mendukung masyarakat[10].

2.2 Analisis dan Perancangan sistem

Analisis sistem adalah menguraikan seluruh sistem menjadi komponen-komponennya guna mengidentifikasi dan mengevaluasi masalah, peluang, hambatan

dan kebutuhan yang diharapkan, sehingga dapat dilakukan perancangan sistem yang baru sesuai kebutuhan[11]. Tahap analisis adalah menganalisis sistem untuk menemukan kelemahan guna memberikan saran perbaikan sistem[12].

Langkah- langkah analisis sistem merupakan sebagai berikut berikut:

1. Identifikasi permasalahannya.
2. Menguasai sistem kerja yang ada.
3. Analisis sistem.
4. Membuat laporan hasil analisis.

Sistem merupakan kumpulan dari subsistem, subsistem tersebut baik yang abstrak maupun fisik yang saling terintegrasi dan bekerjasama untuk mencapai suatu tujuan tertentu. Informasi merupakan data yang diolah, sehingga mempunyai nilai tambah dan berguna untuk pemakai[11]. Sistem informasi merupakan kumpulan dari elemen-elemen yang berinteraksi dan saling berhubungan untuk mencapai tujuan dari sistem. Sistem informasi memberikan nilai tambah bagi proses, produksi, kualitas, manajemen, pengambilan keputusan dan pemecahan masalah, serta keunggulan bersaing, yang tentunya sangat berguna untuk kegiatan bisnis di dalam organisasi[13].

Perancangan Sistem merupakan bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah melalui tahapan analisis. Tujuan dari rancangan sistem ialah buat memenuhi kebutuhan pengguna, dan untuk menyampaikan ilustrasi secara jelas dan desain yang lengkap kepada pemrogram komputer dan pakar teknik lainnya. Tujuan dari perancangan sistem ini diharapkan dapat menciptakan sebuah sistem yang bermanfaat bagi penjualan dan pemasaran di marketplace pertanian, dapat mengatasi permasalahan yang terjadi selama ini[14].

Menurut Kenneth E. Kendall dan Julie E. Kendall, mengemukakan bahwa analisis dan perancangan sistem adalah menganalisis secara sistematis masukan atau aliran data, mengolah atau mentransformasikan data, menyimpan data dan menghasilkan keluaran informasi[15].

2.2.1 *User Interface (UI)*

User Interface atau yang biasa disebut dengan UI adalah tampilan visual suatu produk yang menjadi jembatan antara system dengan pengguna (*user*). Desain dari UI sangat bervariasi, dapat dibuat berdasarkan pada beberapa faktor, seperti tujuan interface, karakteristik pengguna dan karakteristik peringkat interface tertentu. Untuk merancang UI harus didasarkan pada prinsip, yaitu mudah dikenali, tidak membuat kaget pengguna, pemulihan, bantuan dan keberagaman.

Terdapat aturan untuk merancang suatu UI. Delapan aturan emas adalah delapan aturan untuk merancang antarmuka pengguna interaktif (UI) yang mendukung fitur kegunaan.[15]:

1. *Affordance and Visibility*

Pada aturan ini menjelaskan bahwa, tampilan dari fungsi menu harus jelas dan dapat dilihat oleh pengguna sistem akhir. Fungsi sistem juga harus dimanfaatkan secara optimal.

2. *Consistency*

Aturan ini menjelaskan bahwa pengguna yang sering menggunakan aplikasi akan lebih menyukai kecepatan akses ke informasi yang diminta. Jika tingkat interaksi yang diinginkan lebih pendek / pendek dan langsung ditetapkan ke suatu fungsi tanpa melalui aliran menu yang panjang dan banyak kotak dialog.

3. *Shortcut*

Aturan ini menjelaskan bahwa harus ada umpan balik. Umpan balik dimaksudkan untuk memberi informasi yang sejalan dengan tindakan yang akan dilakukan. Pengguna sistem akan tahu tindakan apa yang telah dan akan dilakukan

berkat umpan balik. Umumnya, umpan balik berupa konfirmasi, informasi, atau tindakan..

4. *Feedback*

Pada aturan ini menjelaskan bahwa, harus adanya umpan balik. Umpan balik berfungsi untuk memberikan informasi kepada pengguna sesuai dengan aksi yang dilakukannya. Pengguna system akan mengetahui aksi apa yang telah dan akan dilakukan dengan adanya umpan balik. Umumnya umpan balik berupa konfirmasi, informasi atau suatu aksi.

5. *Dialogs That Yield Closure*

Pada aturan ini menjelaskan bahwa, urutan tindakan sebaiknya diorganisir atau diatur di dalam suatu kelompok bagian awal, tengah dan akhir. Umpan balik yang diberikan akan memberitahukan pengguna sistem bahwa tindakan yang dilakukan sudah benar dan dapat melanjutkan ke tindakan berikutnya.

6. *Error Handling*

Pada aturan ini menjelaskan bahwa, sebuah sistem harus dirancang terlebih dahulu untuk mencegah agar pengguna sistem tidak melakukan kesalahan fatal. Jika terjadi kesalahan yang fatal, maka sistem dapat langsung memberikan pencegahan kesalahan dengan cepat dan memberikan mekanisme yang simple dan mudah dipahami oleh pengguna sistem.

7. *Easy Reversal of Actions*

Pada aturan ini menjelaskan bahwa, sistem dirancang bagi pengguna agar tidak menyulitkan pengguna. Pengguna sistem dibuat untuk tidak takut akan pilihan menu-menu baru karena adanya menu undo atau back dimana memungkinkan pengguna sistem untuk kembali pada tindakan sebelumnya jika salah dalam melakukan tindakan.

8. *Reduce Short-Term Memory Load*

Pada aturan ini menjelaskan bahwa, pengguna tidak disulitkan dengan menu-menu yang banyak di dalam sistem atau aplikasi sehingga pengguna dapat melakukan Tindakan dengan memilih menu yang sederhana tanpa harus mengingat semua perintah atau fungsi menu-menu sistem.

2.2.2 *User Experience (UX)*

User experience merupakan pengalaman yang terkait dengan apa yang dirasakan oleh pengguna dalam menggunakan sistem sehingga mendapatkan kepuasan setelah menggunakannya. Perasaan subjektif pengguna terhadap produk yang mereka gunakan dapat digambarkan pada *User Experience*. Pengguna atau kelompok pengguna yang berbeda mungkin memiliki kesan berbeda mengenai pengalaman penggunaan produk yang sama. Jadi, mengukur pengalaman pengguna biasanya membutuhkan pengumpulan umpan balik dari kelompok pengguna yang lebih besar[16].

User experience memiliki dimensi yaitu sebagai berikut :

a. Kegunaan

Fungsionalitas atau fungsionalitas dicirikan sebagai alat dan fitur yang memberdayakan pengguna untuk menyelesaikan tugas dan menyelesaikannya tujuan mereka. Produk fungsional harus mampu menghasilkan perasaan tertentu pada pengguna.

b. Dapat dipercaya

Kepercayaan merupakan perlindungan privasi yang memungkinkan pengguna untuk memilih bagaimana data pribadinya digunakan. Kepercayaan adalah kesan pengguna atas kemampuan dan pengetahuan penyedia terhadap perilaku yang diharapkan, yang mana adalah integritas.

c. Sosial

Ada hubungan antara produk dan identitas seseorang. Identitas terkait dengan orang lain, dan nilai sosial adalah bagaimana pandangan seseorang di mata orang lain ketika mereka menggunakan suatu produk atau jasa.

d. Kualitas Layanan yang dipersepsikan

Kualitas layanan merupakan ukuran seberapa baik tingkat layanan yang diberikan sesuai dengan harapan pelanggan. Memberikan layanan yang berkualitas berarti memenuhi harapan pelanggan secara konsisten. Kualitas adalah persepsi nyata dari layanan yang diterima pelanggan.

e. Emosional

Nilai emosional adalah utilitas yang diterima dari perasaan atau keadaan afektif yang dihasilkan suatu barang[17].

2.2.3 *Prototype*

Prototype artinya suatu metode dalam pengembangan sistem yang menggunakan pendekatan pengembangan program secara cepat dan bertahap sebagai akibatnya pengguna bisa segera mengevaluasinya.

Manfaat lainnya dari penggunaan prototyping adalah :

1. Mewujudkan sistem nyata merupakan replika dari suatu sistem yang akan berjalan, dengan memperhatikan masukan pengguna untuk memperbaiki sistem tersebut.
2. Pengguna akan lebih bersedia menerima perubahan sistem apa pun yang akan berkembang saat prototipe bekerja hingga hasil pengembangan akhir bekerja nanti.
3. Prototipe dapat ditambahkan atau dikurangi tergantung pada proses pengembangan.
4. Menghemat sumber daya dan waktu dalam menghasilkan produk yang lebih baik dan lebih efisien bagi pengguna.

Langkah pembuatan dalam prototipe adalah sebagai berikut[18]:

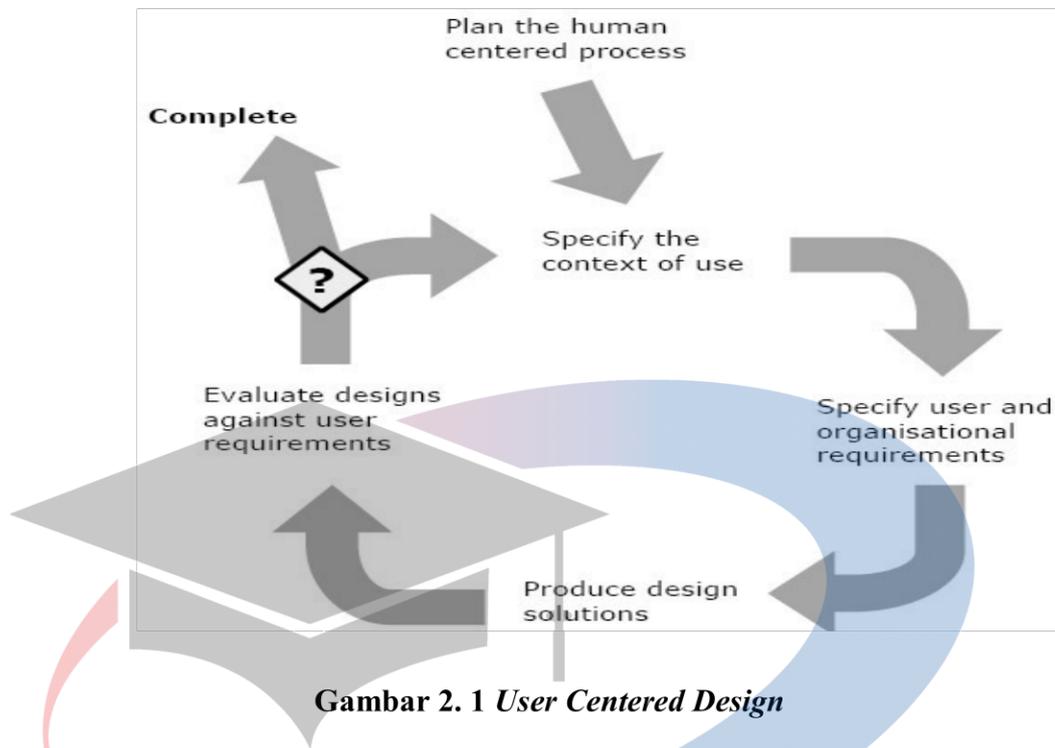
1. Pengumpulan Kebutuhan.
2. Proses desain yang cepat.
3. Membangun prototipe.
4. Evaluasi dan perbaikan.

2.3 *User Centered Design*

User Centered Design (UCD) adalah metode dimana desain proses pengembangan sistem yang berpusat pada pengguna dirancang. Pendekatan UCD melibatkan pengguna pada setiap tahapnya yang merupakan sebuah solusi untuk membangun perangkat lunak yang interaktif. Pendekatan UCD terbukti menghasilkan perangkat lunak dengan tingkat *usability* atau kegunaan yang tinggi. Dimana pengguna dapat memberikan saran mengenai antarmuka (*interface*) aplikasi, sehingga dihasilkan sebuah aplikasi yang memiliki nilai guna tinggi, termasuk kenyamanan penggunaan aplikasi tersebut. Teknik, metode, alat, prosedur, dan proses yang membantu merancang sistem interaktif dibangun dari pengalaman pengguna. UCD menerjemahkan partisipasi dan pengalaman manusia ke dalam proyek [19].

Ada empat proses dalam UCD yakni menurut L. Albani and G. Lombardi[20]:

1. Mengetahui dan mendefinisikan latar belakang pengguna.
2. Menetapkan keinginan pengguna serta organisasi.
3. Solusi perancangan yang dihasilkan.
4. Evaluasi perancangan terhadap kebutuhan pengguna.



Metode yang digunakan dalam UCD adalah dengan melakukan aktivitas sebagai berikut [21]:

1. Kuesioner

Kuesioner ialah salah satu metode pengumpulan informasi yang dicoba dengan metode membagikan seperangkat persoalan ataupun statement tertulis kepada pengguna. Kuesioner jadi efektif apabila telah bisa mengenali dengan tentu variabel yang hendak diukur serta ketahui apa yang dapat diharapkan dari pengguna. Dengan hasil yang diperoleh dari kuesioner, hingga hendak bisa mengenali kebutuhan pengguna.

2. Interview

Interview dilakukan untuk berinteraksi dengan pengguna dengan maksud dengan iktikad buat mencari tahu apa yang diperlukan pengguna dari sistem yang akan dibentuk. Perihal ini dicoba selaku riset pendahuluan untuk menciptakan kasus yang akan diteliti. *Interview* bisa dicoba secara terstruktur serta tidak terstruktur, serta bisa dicoba lewat tatap muka ataupun memakai perlengkapan komunikasi.

3. Task Modelling

Proses menganalisis serta menggambarkan bagaimana pengguna melakukan tugas- tugasnya pada sistem, apa saja yang bisa dicoba serta hal- hal apa saja yang butuh dikenal oleh pengguna. Mengecek tugas- tugas pengguna untuk mengenali dengan baik apa yang diperlukan pengguna dari interface serta gimana pengguna hendak memakainya.

4. *Prototype*

Prototype merupakan sesuatu proses untuk membangun penyelesaian perancangan yang kongkrit berawal dari pengguna serta kebutuhan pengguna. Mewakili model produk yang akan dibentuk, mensimulasikan struktur, fungsionalitas, ataupun pembedahan sistem. Dimungkinkan buat mengimplementasika fungsionalitas yang real. dapat berbentuk *Low- fidelity* ataupun *high- fidelity*. Memberi gambaran tentang totalitas produk ataupun bagian- bagiannya.

2.3.1 Aturan UCD

Aturan-aturan dalam penggunaan metode UCD adalah sebagai berikut[22]:

1. Perspektif

Dalam hal perspektif pengguna akan selalu benar. Jika ada masalah dalam menggunakan sistem, masalahnya ada pada sistem dan bukan pada pengguna.

2. *Instalasi*

Dalam hal ini, pengguna memiliki hak untuk dengan mudah menginstal atau menghapus perangkat lunak dan perangkat keras sistem tanpa konsekuensi negatif apa pun.

3. Pemenuhan

Dalam hal ini, pengguna memiliki hak untuk membuat fungsi sistem persis seperti yang dijanjikan.

4. Instruksi

Dalam hal ini, pengguna memiliki hak untuk dengan mudah mengikuti instruksi (dengan mempertimbangkan manual yang digunakan, bantuan online atau konteks dan pesan kesalahan) untuk memahami dan menggunakan sistem untuk secara efektif mencapai tujuan yang dimaksudkan dan menghindari masalah.

5. Kontrol

Dalam hal ini, pengguna memiliki hak untuk mengontrol sistem dan sistem dapat menanggapi permintaan yang ditransfer dengan benar.

6. Umpan Balik

Dalam hal ini, pengguna berhak atas sistem untuk memberikan informasi yang jelas, mudah dipahami dan akurat tentang tugas yang sedang dilakukan dan kemajuan yang dibuat.

7. Keterkaitan

Dalam hal ini, pengguna memiliki hak untuk mendapatkan informasi yang jelas tentang semua kondisi yang harus dipenuhi oleh sistem untuk mendapatkan hasil yang maksimal.

8. Batasan

Dalam hal ini, pengguna berhak mengetahui batasan kemampuan sistem.

9. *Assistance*

Dalam hal ini, pengguna berhak untuk berkomunikasi dengan penyedia teknologi dan menerima pemikiran atau asumsi yang dapat membantu jika diperlukan.

10. Usability

Dalam hal ini, pengguna dapat menjadi ahli teknologi perangkat lunak dan perangkat keras, dan bukan sebaliknya. Sistem harus dapat digunakan dengan cara yang natural dan intuitif.

2.4 Unified Modeling Language (UML)

UML adalah bahasa spesifikasi baku yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membentuk software. UML adalah metodologi pada berbagai sistem berorientasi objek serta juga adalah alat buat mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam global industry yg adalah baku bahasa pemodelan awam di industri software dan pengembangan sistem[23].

Unified Modeling Language menurut Nugroho, “UML merupakan bahasa pemodelan yang berparadigma pada berorientasi objek”. UML digunakan untuk menyederhanakan masalah kompleks yang dijelaskan melalui diagram [24].

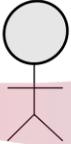
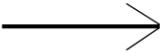
Alat-alat yang digunakan dalam desain berorientasi objek berbasis UML adalah sebagai berikut:

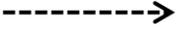
1. *Use case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*

2.4.1 Use Case Diagram

Use Case Diagram terdiri dari antara satu atau lebih aktor dengan sistem informasi yang akan dirancang. *Use case diagram* ialah suatu yang berarti buat memvisualisasikan, menspesifikasikan serta mendokumentasikan kebutuhan sikap sistem. *Use case Diagram* digunakan buat menerangkan aktivitas apa saja yang bisa

dicoba oleh *user*/pengguna sistem yang lagi berjalan[13]. *Use case diagram* memiliki simbol-simbol yaitu sebagai berikut[23]:

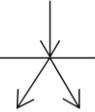
Gambar	Keterangan
	<p>Use case, menjelaskan fungsi yang disediakan oleh sistem sebagai unit yang bertukar pesan antara unit dan partisipan, biasanya dengan menggunakan kata kerja di awal nama <i>use case</i></p>
	<p>Aktor ialah abstraksi dari orang atau sistem lain, yang dapat mengaktifkan fungsi sistem target. Untuk menentukan peserta, perlu ditentukan pembagian kerja dan tugas terkait peran dalam konteks sistem target. Individu atau sistem dapat muncul dalam beberapa peran. Perlu dicatat bahwa peserta terkait dengan <i>use case</i>, tetapi tidak memiliki kendali atas <i>use case</i> tersebut</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang menunjukkan siapa ataupun apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang memakai panah terbuka untuk mengidentikasikan saat aktor berinteraksi secara pasif dengan sistem.</p>

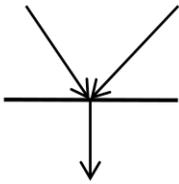
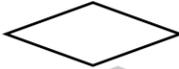
	Include , ialah di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, seperti panggilan ke fungsi program.
	Extend , ialah perpanjangan dari kasus penggunaan lain saat kondisi terpenuhi.

Tabel 2. 1 Simbol *Use Case*

2.4.2 *Activity Diagram* (Diagram Aktivitas)

Activity Diagram menggambarkan alur kerja (*workflow*) atau aktivitas sistem atau proses bisnis. Simbol yang digunakan dalam *activity diagram* yaitu[23] :

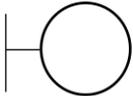
Gambar	Keterangan
	Start point , terletak pada sudut kiri atas dan merupakan awal aktifitas.
	End point , akhir aktivitas
	Activites , menggambarkan suatu proses/aktivitas bisnis.
	Fork (Percabangan), digunakan untuk menunjukkan aktivitas yang dilakukan secara paralel atau untuk menggabungkan dua aktivitas paralel menjadi satu.

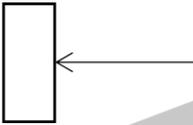
	<p>Join (penggabungan) atau rake, digunakan untuk menunjukkan jadwal.</p>
	<p>Decision Points, menjelaskan pilihan untuk membuat keputusan, benar, salah.</p>
	<p>Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa.</p>

Tabel 2. 2 *Activity Diagram*

2.4.3 *Sequence Diagram* (Diagram Urut)

Interaksi objek yang diatur dalam urutan waktu / peristiwa tertentu dalam suatu proses dapat dijelaskan dengan diagram urutan[13]. Simbol yang digunakan dalam *sequence diagram* yaitu[23] :

Gambar	Keterangan
	<p>Entity Class, merupakan bagian dari sistem yang berisi sekumpulan kelas berupa entity yang membentuk image sistem awal dan menjadi dasar penyusunan database.</p>
	<p>Boundary Class, merupakan berisi kumpulan kelas yang berfungsi sebagai antarmuka atau interaksi antara satu atau lebih pelaku dengan sistem, seperti tampilan formentry dan formulir tercetak.</p>
	<p>Control class, objek yang berisi logika aplikasi yang tidak bertanggung jawab atas entitas,</p>

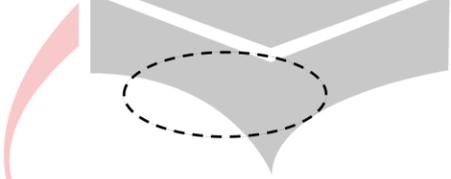
	seperti kalkulasi dan aturan bisnis untuk berbagai objek.
	Message , simbol untuk pengiriman pesan antar kelas.
	Recursive , menjelaskan tentang pengiriman pesan yang dikirim satu sama lain.
	Activation , mewakili pelaksanaan suatu operasi pada suatu objek, panjang bidang ini berbanding lurus dengan durasi operasi.
	Lifeline , garis titik-titik yang terhubung dengan objek, di sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Tabel 2. 3 *Sequence Diagram*

UNIVERSITAS MIKROSKIL

2.4.4 *Class Diagram (Diagram Class)*

Diagram kelas merupakan hubungan antara kelas dan deskripsi rinci dari setiap kelas dalam model desain sistem. Selama analisis, diagram kelas menunjukkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Pada tahap perancangan, diagram kelas berperan dalam menangkap struktur dari semua kelas yang menyusun arsitektur sistem yang dibuat. Berdasarkan kelas yang ada dan fungsi masing-masing[25]. Simbol yang digunakan dalam *Class Diagram* yaitu[26] :

Gambar	Keterangan
	<p>Generalitation, hubungan dimana objek anak (<i>descendent</i>) berbagi hubungan antara perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)</p>
	<p>Class, sekelompok objek yang berbagi properti dan operasi yang sama</p>
	<p>Collaboration, gambaran urutan tindakan yang ditampilkan oleh sistem yang menghasilkan hasil yang terukur bagi actor.</p>
	<p>Realization, operasi sebenarnya dilakukan oleh objek.</p>
	<p>Dependency, hubungan dimana perubahan elemen independen akan mempengaruhi elemen yang bergantung padanya.</p>
	<p>Association, yang menghubungkan antara objek yang lain.</p>
	<p>Nary Association, upaya untuk menghindari asosiasi dengan lebih dari 2 objek.</p>

Tabel 2. 4 *Class Diagram*