

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem merupakan kerangka dari prosedur-prosedur yang berhubungan, yang disusun sesuai skema menyeluruh untuk melaksanakan kegiatan perusahaan. Dalam setiap sistem terdapat proses tertentu untuk menyediakan informasi yang dibutuhkan oleh manajemen dalam pengambilan keputusan, menyediakan informasi yang layak untuk pihak-pihak terkait di luar perusahaan [2].

Sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu [2]:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi sistem yang lebih besar, yang disebut “supra sistem”.

2. Batasan sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan luar sistem (*environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang memengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung sistem (*interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk

subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan *signal (signal input)*. Contoh, di dalam suatu unit sistem komputer. "Program" adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan "data" adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain.

7. Pengolahan Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objektive*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran atau tujuan yang telah direncanakan.

2.1.2 Informasi

Informasi merupakan hasil pengolahan data dengan cara tertentu sehingga lebih berarti dan berguna bagi penerimanya. Sumber dari informasi berupa data yang menggambarkan kejadian secara nyata yang telah terjadi pada saat tertentu [3].

Adapun beberapa jenis-jenis informasi sebagai berikut [3]:

1. *Absolute Information*

Merupakan induk dari informasi yang disampaikan dengan jaminan dan tidak diperlukan penjelasan selanjutnya.

2. *Substitutional Information*

Informasi ini memiliki konsep yang dipakai pada beberapa informasi. Istilah informasi *substitutional* bisa disebut juga komunikasi.

3. *Philosophic Information*

Jenis informasi ini merupakan konsep informasi yang terhubung antara pengetahuan dan kebijakan.

4. *Subjective Information*

Jenis informasi memiliki keterkaitan antara perasaan dan informasi manusia. Informasi ini sangat mendukung penyajinya atau orang yang menyampaikan informasi.

5. *Objective Information*

Jenis informasi tertuju pada informasi-informasi tertentu yang logis.

6. *Cultural Information*

Jenis informasi yang ditekankan pada dimensi *cultural*.

2.1.3 Sistem Informasi

Sistem informasi merupakan elemen-elemen yang terdapat dalam suatu organisasi yang didalamnya terdiri dari sekelompok orang, media, teknologi, prosedur-prosedur serta pengendalian yang dapat digunakan untuk berkomunikasi, transaksi, serta menyediakan informasi dalam pengambilan suatu keputusan [4]. Sistem informasi terdiri dari lima komponen yaitu *hardware* (perangkat keras), *software* (perangkat lunak), data, prosedur, dan manusia [4].

- a. *Hardware* (perangkat keras), mencakup berbagai peralatan fisik seperti komputer dan printer.
- b. *Software* (perangkat lunak), berupa perintah-perintah tertentu yang ditujukan untuk memerintahkan komponen melaksanakan tugasnya.
- c. Data, merupakan komponen paling dasar atau masih mentah dari suatu informasi yang akan diproses lebih lanjut agar dapat berarti dan dapat menghasilkan informasi.
- d. Prosedur, merupakan aturan-aturan yang digunakan untuk menghubungkan berbagai macam perintah dan data untuk menentukan rancangan dan penggunaan sistem informasi.
- e. Manusia, merupakan pelaksana yaitu mereka yang terlibat dalam kegiatan sistem informasi seperti operator, pemimpin dan sebagainya.

2.2 Prototyping

Prototyping merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Dengan metode *prototyping* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi

dalam proses kegiatan pengembangan sistem informasi [5]. Dibuatnya sebuah *prototyping* bagi pengembang sistem bertujuan untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model *prototype* yang dikembangkan, sebab *prototype* menggambarkan versi awal dari sistem untuk kelanjutan sistem sesungguhnya yang lebih besar [5]. Langkah-langkah dalam *prototyping* adalah sebagai berikut [5]:

1. Pengumpulan Kebutuhan

Mengumpulkan kebutuhan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan keseluruhan tujuan dibuatnya perangkat lunak, mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat.

2. Proses Desain yang Cepat

Desain berfokus pada representasi dari aspek perangkat lunak dari sudut pengguna, ini mencakup *input*, proses dan format *output*. Desain cepat mengarah ke pembangunan *prototype*, *prototype* dievaluasi oleh pengguna dan bagian analisis desain dan digunakan untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. *Prototype* diatur untuk memenuhi kebutuhan pengguna, dan pada saat itu pula pengembang memahami secara lebih jelas dan detail apa yang perlu dilakukannya.

3. Membangun *Prototype*

Dalam desain sistem terdapat perancangan relasi dan skema basis data, sebuah relasional skema basis data biasanya dikembangkan dari sebuah *domain Class Diagram* setiap *Class* diidentifikasi secara terpisah. Desain diperlukan dengan tujuan bagaimana sistem akan memenuhi tujuannya dibuat atau diciptakan. Desain sistem terdiri dari kegiatan dalam mendesain yang hasilnya sebuah spesifikasi dari sistem. Bagian dari desain sistem dapat berupa konsep desain *interface*, proses dan data dengan tujuan menghasilkan spesifikasi sistem yang sesuai dengan kebutuhan.

Desain sistem nantinya akan menghasilkan *prototype* paket *software*, dan produk yang baik sebaiknya mencakup [5]:

- a. Fitur menu yang cepat dan mudah.
- b. Tampilan input dan output.
- c. Laporan yang mudah dicetak.
- d. Kamus data yang menyimpan informasi pada setiap *field* termasuk panjang *field*, pengeditan dalam setiap laporan dan format *field* yang digunakan.

e. Basis data dengan format yang sesuai dengan perangkat lunak yang digunakan

4. Evaluasi dan Perbaikan

Setelah *prototype* diterima oleh pengguna tahap selanjutnya adalah evaluasi dan perbaikan *prototype*, tahap ini merupakan tahap akhir dari langkah-langkah *prototyping*. Pada tahapan ini dilakukan evaluasi *prototype* yang telah dibuat dan akan dipergunakan untuk mengetahui apakah *prototype* sudah sesuai dengan kebutuhan atau belum. Jika belum sesuai dengan kebutuhan maka dilakukan perbaikan terhadap *prototype*.

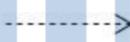
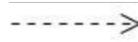
2.3 Teknik Perancangan Sistem

2.3.1 Diagram Use Case

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai [6].

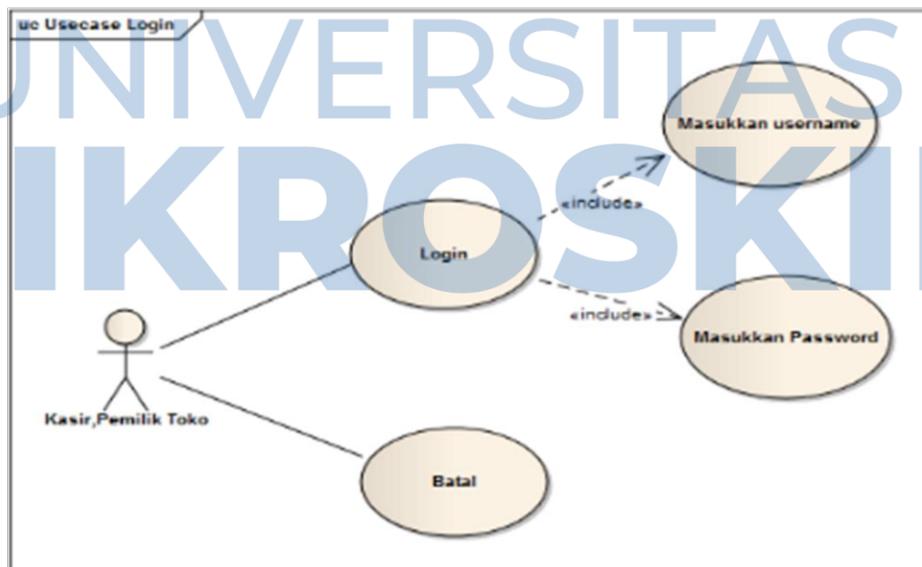
Adapun simbol-simbol Diagram *Use Case* antara lain [6]:

Tabel 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Di bawah ini merupakan contoh gambar *use case diagram* [7]:



Gambar 2.1 *Use Case Login*

Pada gambar *Use Case Login* diatas menjelaskan *actor* pemilik toko/kasir dapat melakukan *login* dengan memilih tombol *login* maka sistem akan menampilkan *textbox* untuk mengisi *username* dan *password*, jika *username* dan *password* benar maka pemilik toko dan kasir dapat mengakses sistem penjualan. Pemilik toko atau kasir memilih tombol batal maka sistem akan membatalkan proses dan keluar dari aktivitas [7].

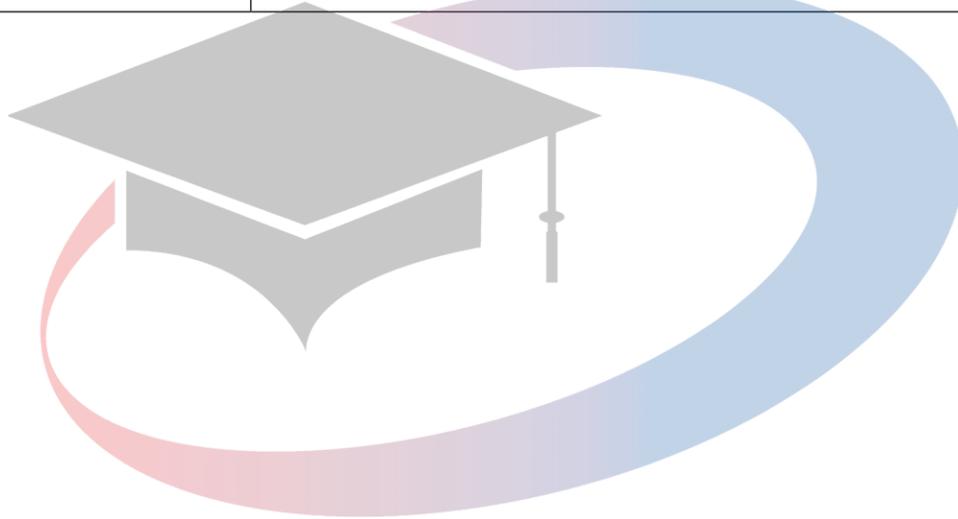
Use Case Description berfungsi melengkapi diagram *use case* agar lebih dipahami konteksnya. Disajikan dalam bentuk tabel, *Use Case Description* adalah gambaran secara general mengenai fungsionalitas proses bisnis berupa skenario yang melibatkan berjalannya suatu sistem. Ada 12 komponen yang terdapat pada *Use Case Description* untuk menjelaskan masing-masing *use case* secara lengkap [8].

Berikut adalah penjelasan dari komponen yang terdapat pada *Use Case Description* tersebut [8]:

Tabel 2.2 Komponen *Use Case Description*

No	Komponen	Deskripsi
1	<i>Use Case ID</i>	Berisikan kode unik untuk membedakan setiap <i>use case</i> .
2	<i>Use Case Name</i>	Berisi nama <i>use case</i> .
3	<i>Description</i>	Berisi penjelasan singkat mengenai fungsi <i>use case</i> .
4	<i>Actor</i>	Berisikan informasi mengenai pengguna atau aktor yang terlibat pada <i>use case</i> tersebut.
5	<i>Pre-Condition</i>	Berisikan kondisi yang harus ada atau sudah terjadi sebelum <i>use case</i> dijalankan. Prasyarat apa yang harus dipenuhi sebelum <i>use case</i> dijalankan.
6	<i>Post Condition</i>	Proses yang dihasilkan dari kegiatan <i>use case</i> .
7	<i>Basic Flow</i>	Berisi penjelasan langkah-langkah normal yang berakhir sukses. Ditandai dengan awal, <i>body</i> dan akhir.
8	<i>Exceptions Flow</i>	Menjelaskan informasi mengenai kendala-kendala yang menyebabkan skenario dasar tidak dapat dipenuhi.
9	<i>Variations</i>	Berisi mengenai tindakan alternatif apabila ada perkecualian dari skenario dasar. Merupakan representasi notasi <i><extend></i> pada diagram <i>use case</i> .

10	Extensions	Berisi mengenai skenario tambahan yang merujuk pada notasi <i><include></i> dan biasanya diidentifikasi apabila ada langkah-langkah yang terlibat tetapi tidak terkait dengan arus konteks skenario awal.
11	<i>Business Rules</i>	Berisi tentang aturan bisnis yang terdapat pada <i>use case</i> .
12	<i>Non-Functional Requirement</i>	Penyampaian kebutuhan diluar fungsi <i>use case</i> dapat disampaikan disini. Umumnya berisi kinerja, kapasitas dan keterbatasan sistem dan lain-lain.



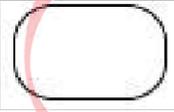
UNIVERSITAS MIKROSKIL

2.3.2 Diagram Activity

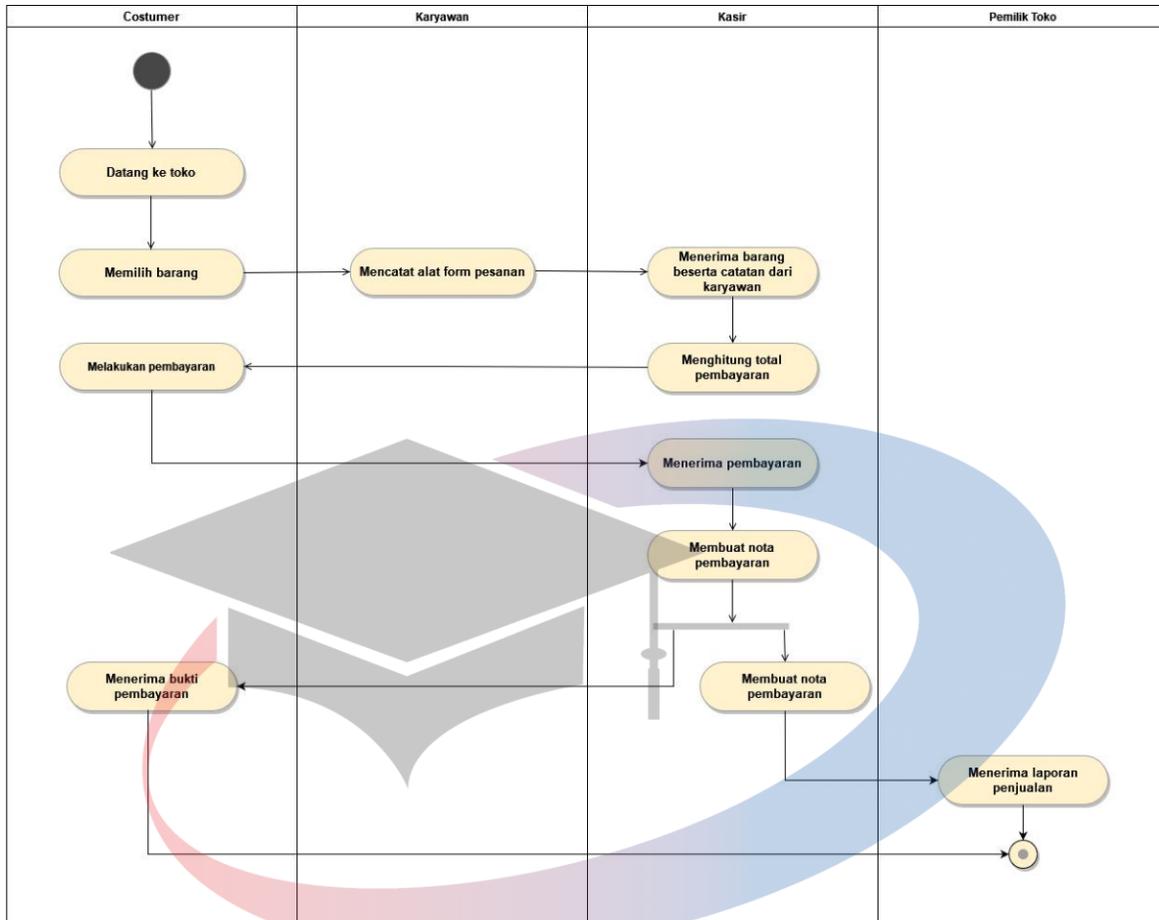
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak [9]. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* memodelkan *event-event* yang terjadi pada use case [10].

Adapun simbol-simbol Diagram *Activity* antara lain [6]:

Tabel 2.3 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Di bawah ini merupakan contoh gambar *activity diagram* [7]:



Gambar 2.2 Activity Diagram

Pada gambar Activity Diagram Sistem Berjalan di atas, diawali customer datang ke toko, kemudian melihat-lihat produk, karyawan menjelaskan detail barang yang ditanyakan, jika customer sudah memilih maka karyawan mencatat pada *form* pemesanan, setelah itu diserahkan ke kasir. Kemudian kasir yang akan melakukan pencatatan total pembelian pada struk/nota pembayaran. Setelah selesai melakukan pembayaran oleh *customer* maka kasir memberikan struk rangkap pertama kepada *customer* dan struk rangkap kedua akan disimpan oleh kasir untuk dibuatkan laporan penjualan. Dimana laporan penjualan tersebut akan diserahkan kepada pemilik toko [7].

2.3.3 Diagram Sequence

Sequence Diagram merupakan UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem, termasuk pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap waktu [9].

Adapun simbol-simbol Diagram *Sequence* antara lain [6]:

Tabel 2.4 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.

2.3.4 Diagram Kelas

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [9]. Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai dengan rancangan yang ada diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron [11].

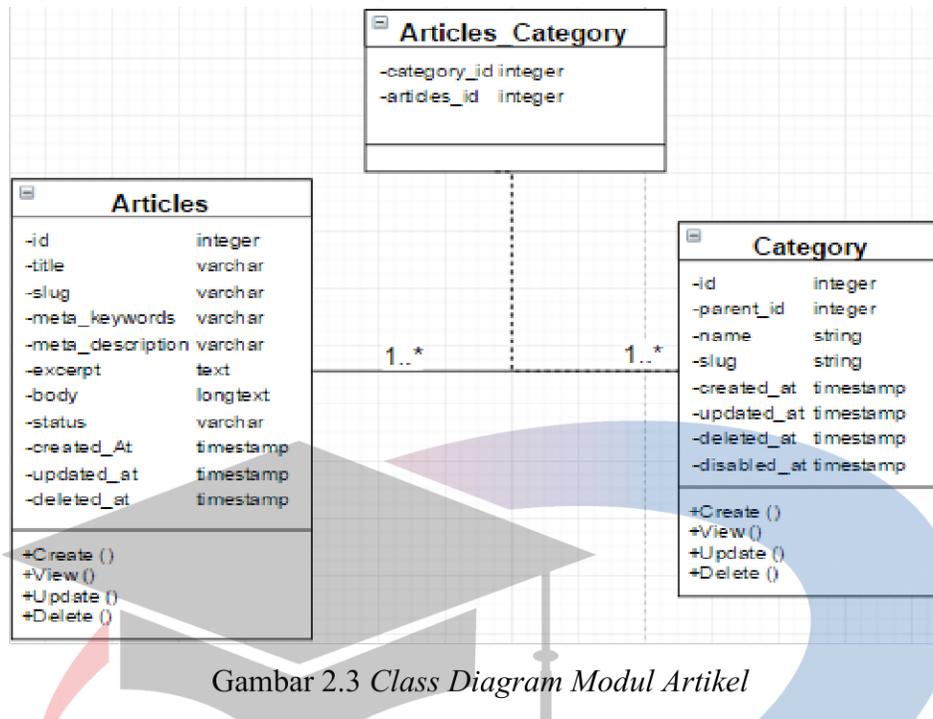
Adapun simbol-simbol Diagram Kelas antara lain [6]:

Tabel 2.5 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Di bawah ini merupakan contoh gambar Diagram Kelas [11]:



Gambar 2.3 Class Diagram Modul Artikel

2.4 Adobe XD

Adobe XD merupakan aplikasi yang dapat digunakan untuk mengembangkan *User Interface* (UI) dan *User Experience* (UX) untuk aplikasi *mobile* maupun web. Aplikasi ini mendukung *wireframing* situs web serta pembuatan *prototype*. Adobe XD CC didesain dan dibuat oleh Adobe System, sehingga mendukung format *file* yang dibuat oleh aplikasi Adobe lainnya. Adobe XD juga mendukung desain gambar serta *vektor*. Fitur utama dari Adobe XD yaitu dapat menyederhanakan desain, memiliki *tools* desain *prototype* yang interaktif, dan *preview prototype* secara *real time* [12].

Adobe XD merupakan perangkat lunak yang digunakan oleh desainer aplikasi web, *mobile*, media desain, dan perancangan UI/UX. Adobe XD dibuat khusus untuk membuat desain UI/UX dan *prototyping* berbagai *platform website*, aplikasi *mobile*, dan lain-lain. *Interface* yang dimiliki Adobe XD cukup sederhana sehingga memudahkan desainer aplikasi dalam mendesain UI. Adobe XD memiliki fitur yang memudahkan pengguna membuat desain UI/UX. Sebuah program yang sangat responsif dan intuitif dengan menggunakan rangka gambar dan membuat *prototype* aplikasi serta situs web. Adobe XD dilengkapi dengan *User experience* (UX) Kits sehingga memudahkan untuk dengan cepat membuat aplikasi asli *platform* Apple iOS [13].

2.5 Pembelian

Pembelian adalah serangkaian tindakan untuk mendapatkan barang dan jasa melalui pertukaran dengan maksud untuk digunakan sendiri atau dijual kembali. Dalam perusahaan dagang pembelian dilakukan dengan menjual kembali tanpa melakukan perubahan bentuk barang, sedangkan pada perusahaan manufaktur pembelian dilakukan dengan merubah bentuk barang yang dibeli dan menjualnya kembali [14].

Fungsi yang terkait pembelian adalah sebagai berikut [14]:

1. Fungsi Gudang
2. Fungsi Pembelian
3. Fungsi Penerimaan
4. Fungsi Akuntansi

Dokumen-dokumen pembelian adalah sebagai berikut [14]:

1. Surat Permintaan Pembelian
2. Surat Permintaan Penawaran Harga
3. Surat Order Pembelian
4. Laporan Penerimaan Barang
5. Surat Perubahan Order Pembelian
6. Bukti Kas Keluar

2.6 Penjualan

Penjualan adalah penyerahan barang dan jasa atau aktivitasnya lainnya dalam periode tertentu dengan membebaskan suatu jumlah tertentu kepada langganan atau konsumen. Kegiatan penjualan merupakan suatu kegiatan yang harus dilakukan oleh perusahaan dengan memasarkan produknya baik berupa barang atau jasa. Kegiatan penjualan yang dilaksanakan oleh perusahaan bertujuan untuk mencapai volume penjualan yang diharapkan dan menguntungkan untuk mencapai laba maksimum bagi perusahaan [15].

Jenis-jenis penjualan adalah sebagai berikut [15]:

- a. *Trade Selling* adalah suatu jenis penjualan yang dilakukan oleh wiraniaga kepada grosir-grosir, dengan tujuan untuk dijual kembali.
- b. *Tehncial selling* adalah berusaha meningkatkan penjualan dengan pemberian saran & nasehat kepada pembeli/konsumen akhir (pembeli/konsumen) dari barang & jasanya. Dalam hal yang

satu ini wirausaha tersebut memiliki tugas utama untuk mengidentifikasi dan juga menganalisis berbagai permasalahan yang dihadapi para pembeli lalu kemudian serta menunjukkan bagaimana produk/jasa yang ditawarkan dapat mengatasi masalah si pembeli/konsumen.

- c. *Missionary Selling* adalah dalam hal yang satu ini wirausaha berusaha meningkatkan penjualan serta dengan mendorong pembeli yang tentunya untuk membeli produk atau jasa dari penyalur perusahaan, dalam hal ini perusahaan tersebut/yang bersangkutan mempunyai penyalur tersendiri dalam pendistribusian produknya/jasanya.
- d. *New Business Selling* adalah berusaha membuka transaksi-transaksi baru dengan cara mengubah calon konsumen menjadi konsumen.

2.7 Persediaan

Persediaan dalam hal ini merupakan suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode tertentu, atau persediaan barang-barang yang masih dalam pengerjaan/proses produksi, ataupun persediaan bahan baku yang menunggu penggunaannya dalam suatu proses produksi. Persediaan meliputi semua barang yang dimiliki perusahaan pada saat tertentu, dengan tujuan untuk dijual kembali atau dikonsumsi dalam siklus operasi normal perusahaan sebagai barang yang dimiliki untuk dijual atau diasumsikan untuk dimasa yang akan datang, semua barang yang berwujud dapat disebut sebagai persediaan/*inventory*, tergantung dari sifat dan jenis usaha perusahaan [16].

Beberapa fungsi persediaan, yaitu [16]:

1. Untuk memberikan suatu stok barang-barang agar dapat memenuhi permintaan yang timbul dari konsumen.
2. Untuk menyesuaikan produksi dengan distribusi. Misalnya, bila permintaan produknya tinggi hanya pada musim panas, suatu perusahaan dapat membentuk stok selama musim dingin, sehingga biaya kekurangan stok dan kehabisan stok dapat dihindari. Demikian pula, bila pasokan suatu perusahaan berfluktuasi, persediaan bahan baku ekstra mungkin diperlukan untuk "menyesuaikan" proses produksinya.
3. Untuk mengambil keuntungan dari potongan jumlah, karena pembelian dalam jumlah besar dapat secara substansial menurunkan biaya produk.
4. Untuk melakukan *hedging* terhadap inflasi dan perubahan harga.

5. Untuk menghindari dari kekurangan stok yang dapat terjadi karena cuaca, kekurangan pasokan, masalah mutu, atau pengiriman yang tidak tepat. "Stok pengaman" misalnya, barang di tangan ekstra, dapat mengurangi risiko kehabisan stok.
6. Untuk menjaga agar operasi dapat berlangsung dengan baik dengan menggunakan "barang-dalam-proses" dalam persediaannya. Hal ini karena perlu waktu untuk memproduksi barang dan karena sepanjang berlangsungnya proses, terkumpul persediaan-persediaan.



UNIVERSITAS
MIKROSKIL