

BAB II

LANDASAN TEORI

2.1. Beras

Beras adalah biji gabah yang bagian kulitnya sudah dipisahkan dengan cara digiling dan disosoh menggunakan alat pengupas dan penggiling serta alat penyosoh (Astawan & Wresdiyati, 2004).

Beras adalah biji-bijian (serealia) dari famili rumput-rumputan (gramine) yang kaya akan karbohidrat sehingga menjadi makanan pokok manusia, pakan ternak dan industri yang mempergunakan karbohidrat sebagai bahan baku. Beras merupakan salah satu makanan pokok (Diniati, 2010).

2.1.1. Jenis-jenis Beras

Menurut (Laseduw, 2014) jenis-jenis beras yang dapat ditemukan dipasaran antara lain:

2.1.1.1. Beras Putih

Beras putih adalah padi yang sudah digiling dan bersih dari bekatul serta kulit arinya sehingga beras yang dihasilkan berwarna putih. Beras putih memiliki sifat pulen namun dari segi nutrisi zat gizinya lebih rendah daripada jenis beras yang lain.

2.1.1.2. Beras Cokelat

Beras cokelat sebenarnya merupakan beras putih yang masih memiliki bekatul serta kulit ari. Bekatul dan

kulit ari memiliki banyak sekali nutrisi, vitamin, mineral dan juga serat. Beras cokelat terkadang sering dianggap sebagai beras merah karena bentuk dan warnanya hampir sama.

2.1.1.3. Beras Merah

Beras merah mudah sekali dikenali dengan warnanya yang kemerahan. Warna merah tersebut berasal dari lapisan bekatul atau aleuron yang mengandung senyawa antosianin, yaitu suatu zat yang membuat beras ini berwarna merah.

2.1.1.4. Beras Hitam

Beras hitam merupakan beras yang langka. Beras hitam sering disebut *forbidden rice*. Beras hitam bukanlah beras ketan hitam, karena keduanya berbeda. Beras ini mengandung senyawa antosianin yang sangat tinggi, sehingga beras yang dihasilkan berwarna hitam atau keunguan. Beras hitam memiliki tekstur agak pera serta kurang cocok untuk dijadikan nasi. Beras hitam yang baik memiliki warna yang hitam mengkilat serta tidak banyak kutu.

2.1.1.5. Beras Ketan Putih

Beras ketan putih banyak digunakan sebagai bahan baku kue, cake, brownies, dan makanan kecil lainnya. Beras

ketan putih berwarna putih, karena mengandung *amilopektin* yang sangat tinggi.

2.1.1.6. Beras Ketan Hitam

Beras ketan hitam tidak memiliki sifat pulen seperti beras ketan putih. Beras ketan hitam umumnya memiliki tekstur agak pera. Sehingga beras ketan hitam sering dijadikan bahan campuran untuk tapai ketan, bubur ketan hitam maupun bahan baku kue tradisional.

2.2. Data

Data adalah fakta yang tidak sedang digunakan pada proses keputusan, biasanya dicatat dan diarsipkan tanpa maksud untuk segera diambil kembali untuk pengambilan keputusan (Wahyudi & Margono, 2009).

Menurut (Yakub, 2012) Sumber data dapat diperoleh dari berbagai sumber untuk memperolehnya. Sumber data diklasifikasikan sebagai sumber data internal, sumber data personal, dan sumber data eksternal.

- a. Data Internal sumbernya adalah orang, produk, layanan, dan proses. Data internal umumnya disimpan dalam basis data perusahaan dan biasanya dapat diakses.
- b. Data Personal, sumber data personal bukan hanya berupa fakta, tetapi dapat juga mencakup konsep, pemikiran dan opini.

- c. Data Eksternal, sumber data eksternal dimulai dari basis data komersial hingga sensor dan satelit. Data ini tersedia di *compact disk*, *flashdisk* atau media lainnya dalam bentuk film, suara gambar, atlas, dan televisi.

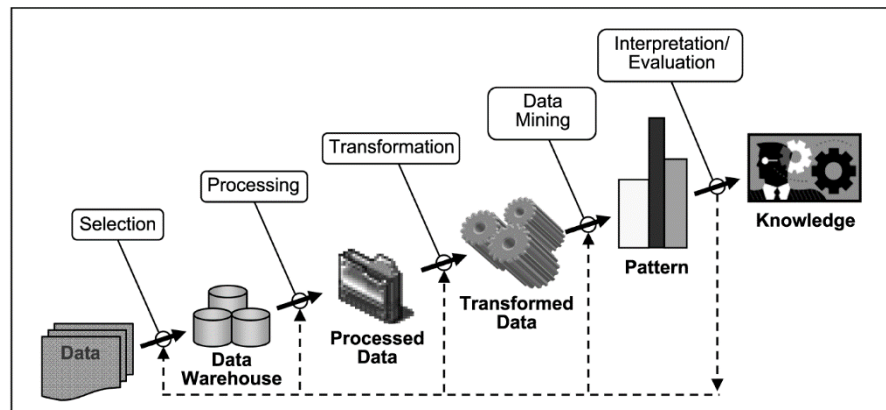
2.3. Data Mining

Menurut (Han, Jiawei, & Kamber, 2006) data mining adalah proses menemukan pola yang menarik dan pengetahuan dari data yang berjumlah besar. Menurut (Connolly & Begg, 2010) data mining adalah suatu proses ekstraksi atau penggalian data yang belum diketahui sebelumnya, namun dapat dipahami dan berguna dari database yang besar serta digunakan untuk membuat suatu keputusan bisnis yang sangat penting.

Dari beberapa teori yang dijabarkan oleh para ahli diatas, bahwa data mining merupakan suatu proses pendukung pengambil keputusan dimana kita mencari pola informasi dalam data.

2.3.1. Tahapan Data Mining

Istilah data mining yang populer saat ini dikenal sebagai *Knowledge Discovery from Data* atau KKD. Data mining merupakan suatu bagian langkah yang penting dalam proses penemuan pengetahuan terutama berkaitan dengan ekstraksi dan perhitungan pola-pola (Han, Jiawei, & Kamber, 2006). Seperti yang ditunjukkan pada gambar berikut ini:



Gambar 2. 1 Tahapan Data Mining

a. *Data Cleaning*

Tahapan ini dilakukan untuk menghilangkan data *noise* dan data yang tidak konsisten dengan tujuan akhir dari proses *data mining*.

b. *Data Integration*

Tahapan ini dilakukan untuk menggabungkan atau mengkombinasikan dari multiple data *source*.

c. *Data Selection*

Pada tahapan ini adalah memilih atau menyeleksi data apa saja yang relevan dan diperlukan dari database.

d. *Data Transformation*

Untuk mentransformasikan data ke dalam bentuk yang lebih sesuai untuk di-*mining*

e. *Data Mining*

Proses terpenting dimana metode tertentu diterapkan dalam database untuk menghasilkan data *pattern*.

f. *Pattern Evaluation*

Untuk mengidentifikasi apakah *interesting patterns* yang didapatkan sudah cukup mewakili *knowledge* berdasarkan perhitungan tertentu

g. *Knowledge Presentation*

Untuk mempresentasikan *knowledge* yang sudah didapatkan dari *user*.

2.3.2. Pengelompokan Data Mining

Menurut (Kusrini & Luthfi, 2009) *data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

a. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan baris data (*record*) lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

c. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

d. Klasifikasi

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

e. Pengklasteran (*Clusterring*)

Pengklasteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas obyek-obyek yang memiliki kemiripan. Klaster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan *record* dalam klaster yang lain. Berbeda dengan klasifikasi, pada pengklasteran tidak ada variabel target. Pengklasteran tidak melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal,

sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

f. Asosiasi

Tugas asosiasi dalam *data mining* adalah untuk menemukan atribut yang muncul dalam satu waktu. Salah satu implementasi dari asosiasi adalah *market basket analysis* atau analisis keranjang belanja.

2.4. Association Rules

Association Rules adalah menampilkan kombinasi atau hubungan diantara item (Zhao, 2013). *Association Rules* meliputi dua tahap yaitu mencari kombinasi yang paling sering terjadi dari suatu *itemset* dan mendefinisikan *condition* dan *result* (untuk *conditional association rules*) (Rindengan, 2012).

2.4.1. Analisis Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam basis data. Nilai *support* sebuah *item* diperoleh dengan menggunakan rumus berikut: (Kusrini & Luthfi, 2009)

$$support(A) = \frac{\text{jumlah transaksi mengandung } A}{\text{total transaksi}} \dots\dots\dots (2.1)$$

Sementara, nilai *support* dari 2 *item* diperoleh dengan menggunakan rumus:

$$\text{support}(A, B) = P(A \cap B)$$

$$\text{support}(A, B) = \frac{\sum \text{transaksi mengandung } A \text{ dan } B}{\sum \text{transaksi}} \dots\dots (2.2)$$

Frequent itemset menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang ditentukan (\emptyset). Misalkan $\emptyset = 2$, maka semua *itemsets* yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent*. Himpunan dari *frequent k-itemset* dilambangkan dengan F_k .

2.4.2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan *asosiatif* $A \rightarrow B$. Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dengan rumus berikut: (Kusrini & Luthfi, 2009)

$$\text{confidence} = P(B|A) = \frac{\sum \text{transaksi mengandung } A \text{ dan } B}{\sum \text{transaksi mengandung } A} \dots (2.3)$$

2.5. Algoritma Apriori

Algoritma Apriori adalah salah satu algoritma yang melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule* (Erwin, 2009). Pada algoritma Apriori menentukan kandidat yang mungkin muncul dengan cara memperhatikan minimum *support* dan minimum *confidence*. *Support* adalah nilai pengunjung atau persentase kombinasi sebuah *item* dalam *database*.

Terdapat dua proses utama pada algoritma apriori (Han, Jiawei, & Kamber, 2011) yaitu sebagai berikut:

a. *Join* (penggabungan)

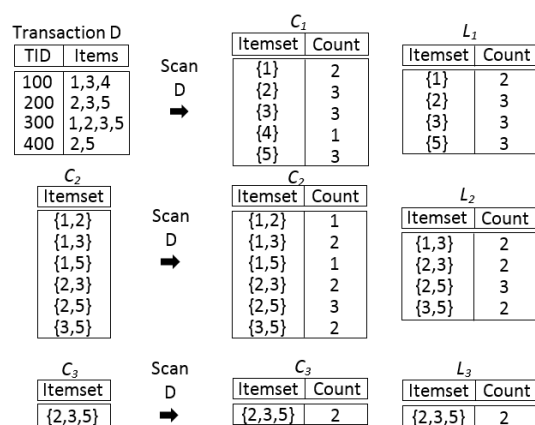
Dalam proses ini, setiap item dikombinasikan dengan *item* yang lainnya sampai tidak terbentuk kombinasi lagi.

b. *Prune* (pemangkasan)

Pada proses ini, hasil kombinasi *item* akan dipangkas dengan menggunakan *minimum support* yang telah ditentukan oleh pengguna.

Algoritma *apriori* bekerja dengan cara menghasilkan kandidat baru dari *k*-itemset pada *frequent itemset* sebelumnya dan menghitung nilai *support k-itemset* tersebut. *Itemset* yang memiliki nilai *support* dibawah dari *minimum support* akan dihapus.

Langkah selanjutnya menghitung *minimum confidence* mengikuti rumus sesuai yang telah ditentukan. *Support* tidak perlu dilihat lagi, karena *generate frequent itemset* didapatkan dari melihat *minsup*-nya. Bila *rule* yang didapatkan memenuhi batasan yang ditentukan dan batasan itu tinggi, maka *rule* tersebut tergolong *strong rules*. Proses perhitungan dalam algoritma berhenti ketika tidak ada lagi *frequent itemset* baru yang dihasilkan.



Gambar 2. 2 Ilustrasi Algoritma Apriori

2.6. Penelitian Terkait

Algoritma Apriori dipilih karena dapat menangani data dalam skala besar. Alasan ini berdasarkan pada jurnal penelitian “Perancangan Aplikasi Prediksi Kecelakaan Lalu Lintas di Percut Sei Tuan dengan Menggunakan Algoritma Apriori” (Harahap, 2013), “Implementasi Data Mining Pada Penjualan Produk Elektronik Dengan Algoritma Apriori” (Pane, 2013) dan “Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat” (Yanto & Khoiriah, 2015).

Tabel 2. 1 : Penelitian Terkait

No.	Nama Peneliti dan Tahun	Judul	Masalah	Hasil
1.	Harahap, 2013	Perancangan Aplikasi Prediksi Kecelakaan Lalu Lintas di Percut Sei Tuan dengan Menggunakan Algoritma Apriori	Belum diketahui faktor penyebab kecelakaan yang paling utama.	Dengan prediksi kecelakaan lalu lintas dapat ditentukan langkah-langkah penanggulangan untuk menurunkan jumlah kecelakaan.
2.	Pane, 2013	Implementasi Data Mining Pada Penjualan Produk Elektronik Dengan Algoritma Apriori	Proses pengolahan data yang kurang efisien.	Menghasilkan olahan data yang efisien sehingga dapat dibuat pola strategi penjualan yang tepat kepada konsumen.
3.	Yanto & Khoiriah, 2015	Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat	Penentuan pola pembelian obat berdasarkan kecenderungan pembelian obat oleh pelanggan.	Ketersediaan obat dan tata letak obat lebih efektif.

2.7. *Software Development Life Cycle (SDLC)*

SDLC merupakan kepanjangan dari *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat sebelumnya. Terdapat banyak model SDLC yang dapat digunakan dalam pengembangan sistem, salah satunya adalah model *waterfall*.

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*) (Mujilan, 2013).

2.8. *UML (Unified Modeling Language)*

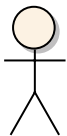
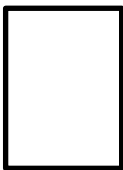
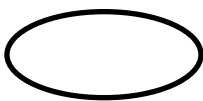

Menurut (Nugroho, 2010) *UML (Unified Modeling Language)* adalah sebuah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma “berorientasi objek”. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

2.8.1. *Use Case Diagram*

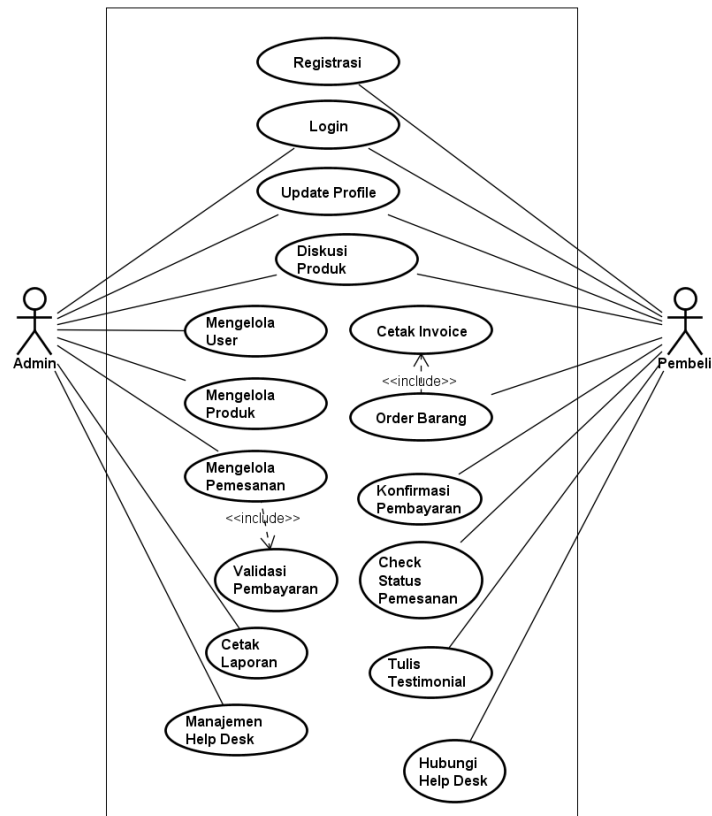
Use Case Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Use*

Case diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (user) mengharapkan interaksi dengan sistem itu. *System Use Case* digunakan untuk memecah atau memilah proses bisnis kedalam interaksi yang berhubungan langsung dengan IT (Supriyono & Prihartanti, 2012).

Tabel 2. 2 : Tabel atribut *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan

Simbol	Nama	Keterangan
		sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
◁-----	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
----->	<i>Extend</i>	Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
_____	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.




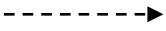
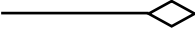


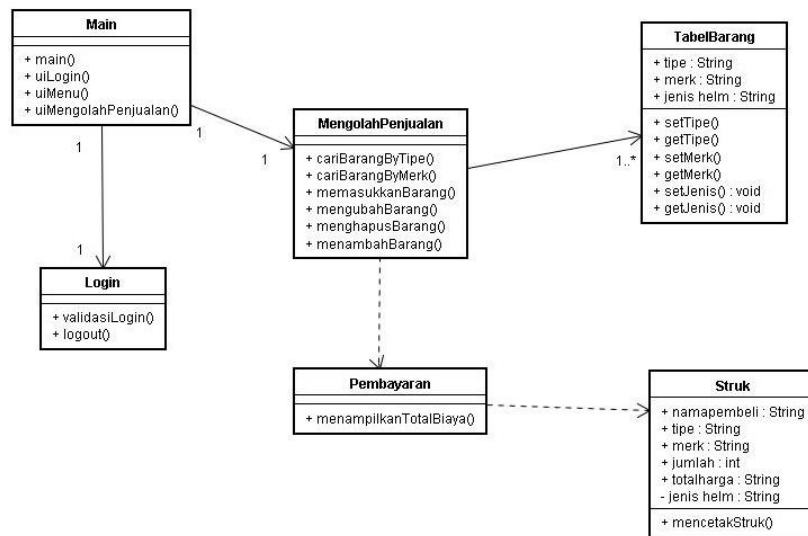
Gambar 2. 3 Contoh Use Case Diagram

2.8.2. Class Diagram

Class diagram merupakan konstruksi atau komponen dari apa yang nantinya dibutuhkan dalam membuat suatu sistem. Sehingga dengan adanya *class diagram* dapat memberikan pandangan *global* atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

Tabel 2. 3 : Tabel atribut Class Diagram

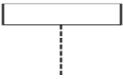
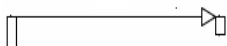
Simbol	Nama	Keterangan
	<i>Association /</i> Asosiasi	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	<i>Directed Association /</i> Asosiasi Berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
	<i>Dependency /</i> Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas.
	<i>Aggregation /</i> Agregasi	Relasi antar kelas dengan makna semua – bagian (<i>whole-part</i>).

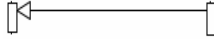
Gambar 2. 4 Contoh *Class Diagram*

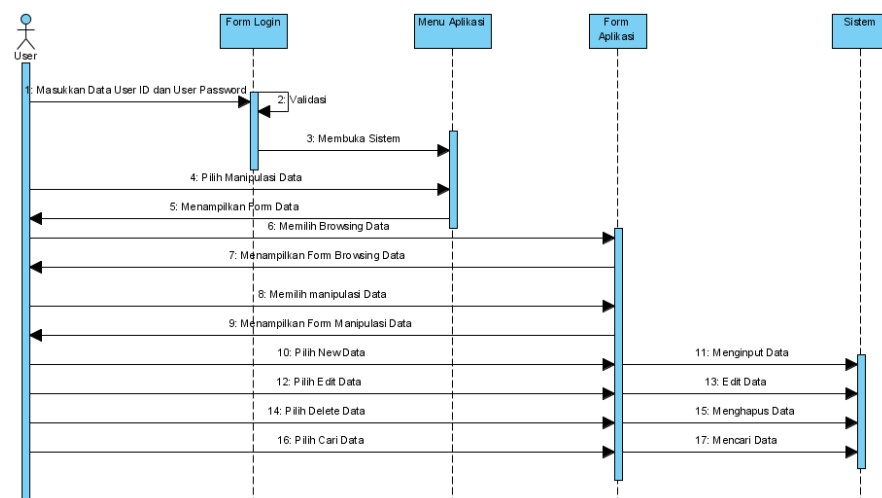
2.8.3. *Sequence Diagram*

Menurut (Indrajani, 2010) *Sequence diagram* merupakan suatu diagram interaksi yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi dan pesan yang ditukar dalam urutan waktu.

Tabel 2. 4 : Tabel atribut *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.






Simbol	Nama	Keterangan
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

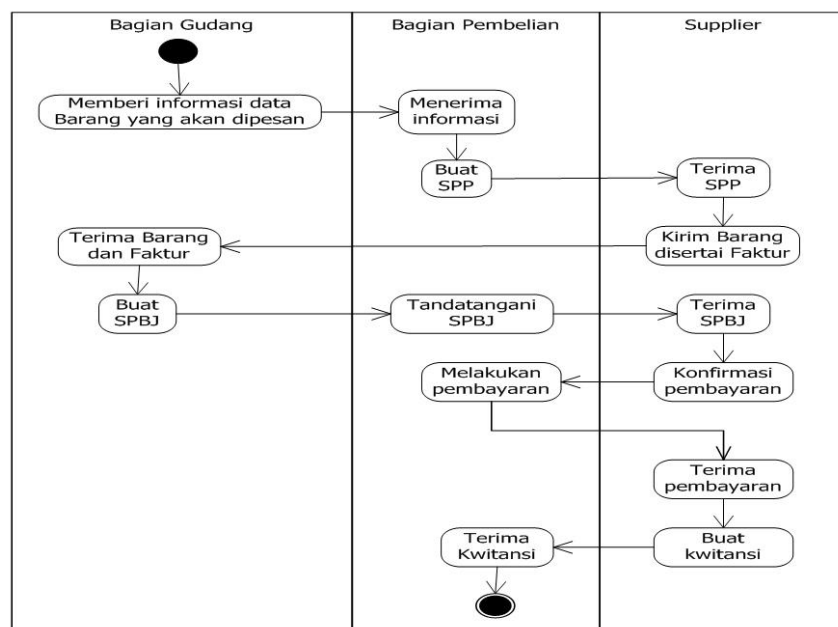
Gambar 2. 5 Contoh *Sequence Diagram*

2.8.4. *Activity Diagram*

Menurut (Indrajani, 2010) *Activity diagram* digunakan untuk menganalisis *behaviour* dengan *use case* yang lebih kompleks dan menunjukkan interaksi-interaksi di antara mereka satu sama lain. *Activity diagram* biasanya digunakan untuk menggambarkan aktivitas bisnis yang lebih kompleks, dimana digambarkan hubungan antara satu *use case* dengan *use case* lainnya.

Tabel 2. 5 : Tabel atribut *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Gambar 2. 6 Contoh *Activity Diagram*