# BAB IILANDASAN TEORI

## **Penelitian Terdahulu**

Penelitian terdahulu ini berguna sebagai panduan pada penelitian skripsi yang dilakukan. Selain menjadi tolak ukur pada penelitian terdahulu ini dapat melihat perbedaan antara penelitian yang telah dilakukan oleh peneliti terdahulu.

### **2.1.1 Peneliti Pertama**

Penelitian yang dilakukan oleh Wicaksono (2019) dengan judul “Sistem Penunjang Keputusan Penentuan Dosen Pembimbing Menggunakan Algoritma Naive Bayes Studi Kasus STIKI Malang”. Proses penentuan dosen pembimbing tugas akhir bagi mahasiswa memerlukan waktu yang lama dan manual. Selain itu Kaprodi harus meneliti judul sesuai dengan calon pembimbing yang mempunyai pengalaman berdasarkan judul yang diangkat. Pada penelitian ini menggunakan metode gaussian naive bayes dan cosine similarity. Sebelum melakukan perhitungan naive bayes melakukan melakukan *text preprocessing* yaitu *case folding, tokenizing, stopword, dan stemming,* Kemudian  melanjutkan dengan melakukan pembobotan TF-IDF. Pada penelitian ini menggunakan PHP sebagai bahasa pemrogramannya. Hasil dari penelitian tersebut adalah sistem dapat mengeluarkan beberapa rekomendasi nama dosen pembimbing dan co pembimbing dari hasil perhitungan metode *naive bayes* dan *cosine similarity.* Kekurangan pada penelitian ini adalah perlu adanya penambahan data *training* agar akurasi atau tingkat keakuratan dosen pembimbing lebih akurat. Hasil persentase akurasi dari penelitian ini yaitu sebesar 81.5%.

### **2.1.2 Peneliti Kedua**

Penelitian yang dilakukan oleh Sabrani (2020) dengan Judul “Metode Multinomial Naive Bayes untuk Klasifikasi Artikel Online Tentang Gempa di Indonesia”. Pada penelitian ini artikel *online* tentang gempa bumi dikelompokkan kedalam 3 kategori yaitu ekonomi, kesehatan, dan pariwisata. Untuk mengelompokkan artikel, pada penelitian ini menggunakan metode multinomial *naive bayes.* Pembobotan tersebut menggunakan teknik TF-IDF. Selain itu sebelum melakukan pembobotan pada penelitian ini menghilangkan *stemming* dan *stopwords removal* pada artikel. Model *naive bayes* yang digunakan pada penelitian ini yaitu *multinomial naive bayes.* Dengan mencari nilai *prior probability* dan *conditional probability.* Hasil klasifikasi menggunakan perhitungan *multinomial naive bayes* pada penelitian ini berhasil melakukan klasifikasi berdasarkan kategori artikelnya. Sedangkan hasil persentase akurasinya yaitu sebesar 95.20%.

### **2.1.3 Peneliti Ketiga**

Penelitian yang dilakukan oleh Rahman (2017) dengan judul “Online News Classification Using Multinomial Naive Bayes”.  Pada penelitian ini klasifikasi yang digunakan yaitu *naive bayes* dengan model *multinomial.* Tahap *text preprocessing* pada penelitian ini yaitu *case folding, tokenization, stemming* dan *filtering.* Sedangkan untuk pembobotan dalam penelitian menggunakan 2 macam fitur seleksi yaitu menggunakan *DF-Thresholding* dan TF-IDF. Dari hasil penelitian ini dapat disimpulkan bahwa metode *multinomial naive bayes* dapat digunakan untuk klasifikasi berita. Metode *multinomial naive bayes* dengan menggunakan pembobotan TF-IDF memiliki hasil yang lebih baik daripada metode *multinomial naive bayes* dengan menggunakan fitur seleksi *DF-Thresholding.* Pada penelitian ini didapat dua macam hasil akurasi. Hasil dari persentase akurasi pertama menggunakan TF-IDF yaitu sebesar 94.29%, sedangkan hasil persentase akurasi kedua menggunakan *DF-Thresholding* yaitu sebesar 86.62%.

### **2.1.4 Peneliti Keempat**

Penelitian yang dilakukan oleh Ilyas (2015) dengan judul “Ekstraksi Informasi 5W1H pada Berita *Online* Bahasa Indonesia”. Kesulitan dalam pencarian informasi pada portal berita dikarenakan informasi penting tertutup oleh informasi yang tidak dibutuhkan. Informasi utama dapat dikategorikan 5W1H yang merupakan informasi inti dari sebuah berita. Sehingga agar mendapatkan informasi utama, pada penelitian ini menggunakan teknik Ekstraksi Informasi untuk mengekstraksi informasi lebih dari satu berita. Proses pengenalan entitas dapat dilakukan untuk dokumen berita menghasilkan dokumen terstruktur dengan kelas seperti nama, seseorang, organisasi, lokasi/tempat, tanggal atau uang. Pada penelitian ini kelas yang dituju adalah 5W1H. Pada penelitian tersebut ekstraksi 5W1H masih perlu dikembangkan untuk mendapatkan kinerja terbaik. Dengan memperbaiki kinerja model, dan memanfaatkan model ekstraksi 5W1H untuk peringkasan berita.

### **2.1.5 Peneliti Kelima**

Penelitian yang dilakukan oleh Rahutomo (2019) dengan judul “Eksperimen *Naive Bayes* Pada Deteksi Berita *Hoax* Berbahasa Indonesia”. Untuk klasifikasi mendeteksi berita yang bersifat valid atau palsu berbahasa Indonesia. Sebelum melakukan klasifikasi data latih dan uji pada penelitian ini melakukan tahap *preprocessing* yang terdiri dari *case folding, tokenizing, stopword removal,* dan *term frequency.* Selain menggunakan metode *naive bayes* untuk mengklasifikasikan berita, pada penelitian ini menggunakan komponen PHP-ML atau PHP-Machine Learning. Penelitian ini menggunakan library PHP karena bersifat *open source* sehingga *developer* dapat menggunakan sesuai dengan kebutuhan. Pada penelitian kelima ini hasil dari presentase akurasi sebesar 82.6%.

### **2.1.6 Peneliti Keenam**

Penelitian yang dilakukan oleh Sari (2019) dengan judul “Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Pada Wisata TMII Berbasis Website ”. Pada penelitian ini menggunakan opini dari wisatawan dengan mengambil data ulasan dari Tripadvisor yang kemudian diklasifikasikan kedalam kalimat negatif atau positif. Seperti pada penelitian sebelumnya, pada penelitian ini melakukan *text preprocessing* terlebih dahulu yang terdiri dari *tokenizing, stopword removal,* dan *stemming.* Selanjutnya dilanjutkan dengan melakukan klasifikasi menggunakan metode *naive bayes.* Pada penelitian ini untuk membangun sistemnya, menggunakan bahasa pemrograman PHP dan HTML. Hasil dari persentase akurasinya yaitu sebesar 70%.

 Tabel 2. 1 Ringkasan hasil penelitian terdahulu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Penulis** | ***Text Preprocessing*** | ***Selection Data*** | **Algoritma Klasifikasi** | ***Tool*** | **Hasil** |
| 1 | Wicaksono (2019) | * *Case Folding*
* *Tokenizing*
* *Filtering/Stopword*
* *Stemming*
 | * TF-IDF
* *Cosine Similarity*
 | *Gaussian Naive Bayes* | PHP dan MySQL | Persentase akurasi sebesar 81.5% |
| 2 | Sabrani (2020) | * *Tokenizing*
* *Stemming*
* *Stopword removal*
 | * TF-IDF
 | *Multinomial Naive Bayes* | tidak dijelaskan | Persentase akurasi sebesar 95.20% |
| 3 | Rahman (2017) | * *Case Folding*
* *Tokenizing*
* *Stemming*
* *Filtering Stopword*
 | * *DF-Thresholding*
* TF-IDF
 | *Multinomial Naive Bayes* | tidak dijelaskan | * Persentase akurasi mengunakan TF-IDF 94.29%
* Persentase akurasi mengunakan *DF-Thresholding* 86.62%
 |
| 4 | Ilyas (2015) | * *Tokenisasi*
* Pemberian Fitur
 | * tidak dijelaskan
 | * *SVM*
* *IBK*
* *ID3*
 | tidak dijelaskan | tidak dijelaskan |
| 5 | Rahutomo (2019) | * *Case Folding*
* *Tokenizing*
* *Stopword Removal*
 | * TF-IDF
 | *Gaussian Naive Bayes* | PHP-Machine Learning dan MySQL | Persentase akurasi sebesar 82.6% |
| 6 | Sari (2019) | * *Tokenizing*
* *Stopword removal*
* *Stemming*
 | * tidak dijelaskan
 | *Gaussian Naive Bayes* | PHP dan HTML | Persentase akurasi sebesar 70% |

## **PT. Assist Software Indonesia Pratama**

Assist Software Indonesia (Assistindo) merupakan perusahaan *software house* agensi. Didirikan pada bulan Juni 2000 oleh beberapa personil yang telah berkompeten pada masing-masing bidangnya. Mempunyai visi dan misi yang sama tentang bagaimana sebaiknya membina suatu dunia bisnis dengan dukungan software yang mencakup semua aspek penting dalam memajukan bisnis. Berawal dari motto *High Performance, Easy to Used*. Assist Software Indonesia (Assistindo), menciptakan solusi-solusi terpadu pada sebuah software atau aplikasi menurut lingkup tertentu (produk) maupun yang dapat disesuaikan (proyek) dengan kebutuhan dunia usaha yang berkembang saat ini. Produk yang dikeluarkan bergerak di bidang perbankan (bpr, bmt, koperasi, dan lain-lain), retail, tekstil, hrd, dan sebagainya. Sistem yang mulanya dirancang untuk mendukung operasi yang tersentralisasi ataupun tidak tersentralisasi akan ditingkatkan untuk memampukan perusahaan induk dan cabangnya beroperasi sebagai sebuah koordinat suatu sistem yang terintegrasi. Struktur organisasi PT. Assist Software Indonesia Pratama ditunjukkan pada gambar 2.1.



Gambar 2.1 Struktur Organisasi

## ***Text Mining***

Secara umum *text mining* merupakan proses menganalisis dan mengeksplorasi sejumlah data yang besar dimana biasanya datanya didapat dari dokumen atau dari data teks yang tidak terstruktur. Selain bertujuan untuk mendapatkan informasi dari sebuah dokumen, *text mining* bertujuan untuk mengidentifikasi konsep, ekstraksi pola, kata kunci, topik, dan sebagainya. Menurut Wijaya dan Santoso (2016) *text mining* adalah teori tentang pengolahan koleksi dokumen dalam jumlah besar yang ada dari waktu ke waktu dengan menggunakan beberapa analisis, tujuan dari pengolahan teks tersebut adalah untuk mengetahui dan mengekstraksi informasi yang berguna dari sumber data dengan identifikasi dan eksplorasi pola menarik dalam kasus *text mining*, sumber data yang diperlukan adalah kumpulan atau koleksi dokumen tidak terstruktur dan memerlukan adanya pengelompokan untuk diketahui informasi sejenis.

*Text mining* adalah bidang khusus dari data mining, yang membedakan adalah jenis datasetnya. Dataset yang dipergunakan dalam data mining seperti data-data terstruktur dalam database, sementara pada *text mining* data yang dipergunakan adalah dataset yang tidak terstruktur berupa teks misalnya Word dokumen, PDF file, XML file, dll. Dataset merupakan kumpulan dari object data yang digambarkan dengan sejumlah atribut (sifat atau karakteristik) karakter dasar object data. Misalnya berat badan yang memberikan nilai kuantitatif dari berat badan seseorang.

## ***Text Preprocessing***

Menurut Sabrani (2020) *Text preprocessing* merupakan proses untuk mentransformasikan teks ke dalam kumpulan kata. Teks merupakan data yang tidak terstruktur, yang mana cukup sulit untuk diproses dengan komputer. Operasi numerik pun tidak dapat diaplikasikan pada data teks. Oleh karena itu, perlu dilakukan *preprocessing* pada teks untuk mendapatkan data yang dapat diolah menggunakan komputer.Berikut ini langkah-langkah untuk *preprocessing* yaitu (Wicaksono, 2019) :

1. *Case Folding*

Pada tahap ini mengubah semua kata menjadi huruf kecil atau *lower case*. Pada tahap ini hanya menerima karakter huruf saja. Selain huruf ‘a’ sampai dengan ‘z’, maka akan dianggap sebagai delimiter (pembatas) dan dihilangkan.

1. *Tokenizing*

*Tokenizing* yaitu proses memisahkan kata atau pemecahan aliran konteks tekstual menjadi istilah atau kata-kata.

1. *Filtering*

Filter *stopword* digunakan untuk menghilangkan kata-kata yang dianggap tidak memiliki makna atau kata-kata yang sering muncul pada suatu dokumen dan tidak memiliki pengaruh apapun terhadap ekstraksi klasifikasi teks sebab kata-kata yang dihilangkan tersebut tidak diukur sebagai kata kunci. Contohnya adalah kata penunjuk waktu, kata tanya, kata sambung, seperti di, ke, kemudian, atau, lalu, adalah, dan lain sebagainya. Suatu teknik pengeliminasian *stopword* yang efisien dibutuhkan pada pengaplikasian pemrosesan bahasa seperti normalisasi ejaan. dan *stemming.* Cara umum yang digunakan dalam menentukan kata sebagai *stopword* adalah dengan menghitung frekuensi kemunculan pada suatu dokumen. Kata yang memiliki frekuensi terbesar biasanya dianggap sebagai *stopword.*

1. *Stemming*

Merubah bentuk kata menjadi kata dasar (*stem*). Proses *stemming* pada kata Bahasa Indonesia berbeda dengan *stemming* pada kata Bahasa Inggris. Proses *stemming* pada kata Bahasa Inggris merupakan proses untuk mengeliminasi sufiks pada kata sementara proses *stemming* pada Bahasa Indonesia adalah proses untuk mengeliminasi sufiks, prefiks, dan konfiks.

## ***Information Extraction* (IE)**

Ekstraksi informasi digunakan untuk mendapatkan informasi yang terstruktur seperti entitas, hubungan antar entitas, dan atriasbut yang menggambarkan entitas dari sumber yang tidak terstruktur. Ekstraksi informasi berperan sebagai sistem yang akan mengenali data yang tidak terstruktur yang memiliki informasi yang belum dikategorikan dan belum memiliki arti yang spesifik. Tidak terstruktur bukan berarti secara data yang membingungkan atau tidak jelas dengan artian bahwa informasi yang terkandung didalamnya tidak dapat langsung diterjemahkan oleh komputer sehingga membutuhkan sistem yang dinamakan ekstraksi informasi.

Pendekatan yang paling umum dalam melakukan ekstraksi informasi adalah dengan penggalian teks/*text mining.* Penggalian teks didefinisikan sebagai cara melakukan penemuan otomatis informasi yang tidak ditemukan sebelumnya, dilakukan oleh komputer secara otomatis dan diambil dari sumber informasi yang berbeda (Hearst, 2004) dalam Ilyas (2015).

## ***Term Frequency - Inverse Document Frequency* (TF-IDF)**

Umumnya TF-IDF digunakan sebagai faktor untuk menghitung bobot pada pengambilan informasi. Nilai TF-IDF meningkat setiap banyaknya kata muncul pada dokumen, namun turun apabila frekuensi kata sering muncul, hal ini untuk menangani kata-kata yang sering muncul. (Cahyono, 2017).

(2. 1)

*Term Frequency* (TF) adalah proses pemberian bobot pada *term* atau kata berdasarkan jumlah kemunculan kata dalam dokumen. TF dapat ditunjukkan dengan 3 cara (Salton & Buckley, 1998) dalam (Cahyono, 2017), yaitu:

1. Menggunakan nilai biner, yaitu diberi nilai 1 untuk kata-kata yang terdapat pada dokumen dan nilai 0 terhadap kata-kata yang tidak muncul pada dokumen. Pada konsep ini, frekuensi kemunculan kata tidak dimasukkan ke dalam perhitungan.
2. Menggunakan nilai frekuensi kemunculan kata secara langsung untuk menjadi TF.
3. Menggunakan nilai pecahan term yang telah dilakukan normalisasi

*Document Frequency* (DF) adalah banyaknya kata pada semua dokumen yang ada. Untuk menghitung *inverse document frequency* (IDF) dapat menggunakan rumus sebagai berikut (Salton & Buckley, 1998) dalam (Cahyono, 2017):

$$IDF\left(t,d\right)=log⁡(\frac{N}{Df\left(t,d\right)})$$

Keterangan:

N = Jumlah dokumen

Df(t,d) = Banyak dokumen (d) dalam kumpulan dokumen D yang mengandung *term* (t)*.*

Sedangkan untuk menghitung TF-IDF dapat diambil dari nilai TF dan nilai IDF dengan rumus:

$$TF-IDF(t,d,D)=TF\left(t,d\right) x IDF(t,d)$$

## **Klasifikasi**

Menurut Siregar dan Puspabhuana (2017) Klasifikasi adalah proses untuk menentukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu object yang labelnya tidak diketahui. Dalam proses klasifikasi biasanya dibagi menjadi dua fase, yaitu *learning* dan *test.* Pada fase *learning,* sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model perkiraan. Sedangkan pada fase test model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi model ini dapat dipakai untuk prediksi kelas data yang belum diketahui.

Mengklasifikasi dokumen merupakan salah satu cara untuk mengorganisasikan dokumen. Dokumen-dokumen yang memiliki isi yang sama akan dikelompokkan ke dalam kategori yang sama. Dengan demikian, orang-orang yang melakukan pencarian informasi dapat dengan mudah melewatkan kategori yang tidak relevan dengan informasi yang dicari atau yang tidak menarik perhatian (Feldman, 2004) dalam (Sianturi, 2017). Tujuan dari klasifikasi dokumen yaitu untuk mengelompokkan dokumen yang terstruktur atau tidak terstruktur ke dalam kelompok-kelompok yang menggambarkan isi dari dokumen tersebut.

## ***Naive Bayes***

Secara umum *naive bayes* digunakan untuk mengklasifikasikan sekumpulan dokumen. Menurut Siregar dan Puspabhuana (2017) *Naive Bayes* yaitu mengklasifikasikan dokumen menggunakan metode probabilitas dan statistik (memperkirakan peluang di masa depan berdasarkan pengalaman di masa lampau) yang dikemukakan oleh ilmuwan inggris Thomas Bayes sehingga dikenal sebagai *teorema Bayes.* Pada *teorema* ini dikombinasikan dengan “*naive*” dimana diasumsikan kondisi antar atribut saling bebas. Sehingga keberadaan antara atribut tidak ada kaitannya satu sama lain.

### **2.8.1 Multinomial Naive Bayes Classifier**

Menurut Sabrani (2020) *Multinomial naive bayes* termasuk metode *supervised learning,* sehingga setiap data perlu diberikan label sebelum dilakukan *training.* Klasifikasi dokumen bertujuan untuk menentukan kelas terbaik untuk suatu dokumen. Kelas terbaik dalam klasifikasi *naive bayes* ditentukan dengan mencari *maximum a posteriori* (MAP) kelas $C\_{map}$ melalui persamaan.

$$C\_{map}=arg max\_{c\in C}\hat{P}(c)\prod\_{k=1}^{n}\hat{P}(t\_{k}|c)$$

Keterangan:

*arg max* = Fungsi untuk mencari nilai *posterior probability* terbesar suatu kelas

$\hat{P}(t\_{k}|c) $= *Conditional probability,* peluang kemunculan kata *k* dalam suatu kelas tertentu

$\hat{P}(c)$ = *Prior probability,* peluang kemunculan sebuah kelas dari seluruh pengamatan yang dilakukan.

Adapun menghitung nilai $\hat{P}(c)$ nilai probabilitas sebuah dokumen d berada dalam kelas c dinyatakan dengan rumus:

$$\hat{P}\left(c\right)= \frac{N\_{c}}{N}$$

Keterangan:

$\hat{P}(c)$ = *Prior probability* suatu dokumen berada di kelas c

$N\_{c}$ = Jumlah dokumen di kelas c

N = Jumlah seluruh dokumen *training*

 Sedangkan untuk menghitung *conditional probability* dengan menggunakan pembobotan TF-IDF dapat menggunakan rumus dibawah ini:

$$\hat{P}\left(t|c\right)= \frac{W\_{ct}+1}{(\sum\_{W'\in V}^{}W\_{ct'}+B'}$$

Keterangan:

$W\_{ct}$ = Bobot TF-IDF *term t* pada dokumen dengan kategori c

 $\sum\_{W'\in V}^{}W\_{ct'}$= Jumlah bobot TF-IDF seluruh *term* pada kelas c

*B’* =Jumlah IDF seluruh *term* pada *vocabulary*

## **CodeIgniter**

Menurut Betha Sidik (2012) dalam Destiningrum (2017) CodeIgniter adalah “Sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal”.

Framework Codeigniter dikembangkan oleh Rick Ellis, CEO Ellislab, Inc. Kelebihan dari framework codeigniter jika dibandingkan dengan framework lain adalah sebagai berikut (Destiningrum, 2017):

1. Gratis (Open-Source)

Kerangka kerja Codeigniter memiliki lisensi dibawah Apache/BSD open-source sehingga bersifat bebas atau gratis.

1. Berukuran kecil

Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan framework lain yang berukuran besar dan membutuhkan resource yang besar dan juga dalam eksekusi maupun penyimpanannya.

1. Menggunakan konsep M-V-C

Codeigniter merupakan konsep M-V-C (Model-View-Controller) yang memungkinkan pemisahan antara layer application-logic dan presentation. Dengan konsep ini kode PHP, query Mysql, Javascript dan CSS dapat saling dipisah-pisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau maintenance.

1. Model Kode merupakan program (berupa OOP *class*) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (input-edit-delete).
2. View Merupakan kode program berupa template atau PHP untuk menampilkan data pada browser.
3. Controller merupakan Kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan view.

## **MySql**

Menurut Adi Nugroho (2011) dalam (Destiningrum, 2017) MySQL (*My Structured Query Language*) adalah “Suatu sistem basis data relation atau Relational Database management System (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi multi user (banyak pengguna). MySQL didistribusikan gratis dibawah lisensi GPL (*General Public License*). Dimana setiap program bebas menggunakan MySQL namun tidak bisa dijadikan produk turunan yang dijadikan closed source atau komersial”.

## ***Unified Modeling Language* (UML)**

 Menurut Windu Gata, Grace (2013:4) dalam Hendini (2016), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang diperlukan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah *Use Case Diagram,* Diagram Aktivitas (*Activity Diagram*), Diagram Urutan(*Sequence Diagram*), dan Diagram Kelas (*Class Diagram*)*.*

## **2.12 *Activity Diagram***

*Activity Diagram* menggambarkan *workflow* (alur kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel 2.5 (Hendini, 2016).

Tabel 2. 2 Tabel *Activity Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Keterangan** |
|  | *Start Point,* diletakkan pada pojok kiri atas dan merupakan awal aktivitas |
|  | *End Point,* akhir dari aktivitas |
|  | *Activities,* menggambarkan suatu proses/kegiatan bisnis |
|  | *Frok/*percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu |
|  | *Join* (penggabungan) atau *rake,* digunakan untuk menunjukkan adanya dekomposisi |
|  | *Decision Points,* menggambarkan pilihan untuk pengambilan keputusan yaitu *true* atau *false* |
|  | *Swimlane,* pembagian *activity diagram* untuk menunjukkan siapa melakukan apa |