**BAB II**

**TEORI DASAR**

Pada bab ini akan dijelaskan mengenai iPhone, Xcode sebagai Software Development Kit yang dipergunakan pada Tugas Akhir ini dan struktrur dari WordNet sebagai sumber Lexical Database. Penjelasan dari iPhone akan mencakup perkembangan dari mobile device ini, dan juga fitur-fitur yang membuat iPhone memiliki nilai lebih tersendiri dibanding dengan mobile device lain. Penjelasan mengenai Xcode akan meliputi penjelasan dari perkembangan iPhone SDK dan fitur-fitur didalamnya. Sedangkan penjelasan mengenai WordNet akan meliputi struktur dari WordNet serta struktur penyimpanan data pada index dan data file.

* 1. **iPhone**

iPhone adalah sebuah mobile device yang dikembangkan dan dijual oleh Apple, Inc. Dalam perkembangannya iPhone membawa dampak perubahan yang besar dalam perkembangan mobile device. Fitur-fitur yang dimiliki oleh iPhone menaikkan standar fungsionalitas yang harus dimiliki oleh sebuah mobile device. Pada subbab ini akan dijelaskan mengenai perkembangan iPhone dan fitur-fitur yang dimilikinya.

1. **Pengenalan iPhone**

iPhone adalah mobile device yang dikenalkan pada publik pada Januari 2007, dan pertama kali dijual ke publik pada bulan Juli 2007. Sampai saat ini sudah ada 4 generasi iPhone yang diluncurkan oleh Apple, Inc. yaitu iPhone 2G yang diluncurkan pada tahun 2007, iPhone 3G yang diluncurkan pada tahun 2008, iPhone 3GS yang diluncurkan pada tahun 2009, dan yang terakhir iPhone 4 yang diluncurkan pada tahun 2010.

Apple memperkenalkan iPhone sebagai alat 3-in-1. Fitur-fitur utama yang dikenalkan oleh Apple tersebut adalah fitur telepon, iPod dan internet device. Seperti mobile device pada umumnya, iPhone dapat melakukan dan menerima panggilan telepon. Disamping itu iPhone juga dapat digunakan sebagai pemutar musik dan video dengan memanfaatkan aplikasi iPod yang ada didalamnya. Sebagai tambahan, iPod adalah alat pemutar music berformat .mp3 milik Apple yang sangat terkenal. iPhone juga memiliki HTML browser yang bernama Safari. Para pengguna iPhone juga dapat menambahkan aplikasi-aplikasi yang dibutuhkan ke dalam iPhone dengan mengunduhnya melalui AppStore. Aplikasi tersebut dapat berupa permainan, kamus, gps, dan berbagai jenis aplikasi lain.



**Gambar 2.1**

**iPhone 4**

Generasi iPhone yang pertama, yaitu iPhone 2G masih memiliki fitur-fitur yang sangat minimal. Setahun berikutnya diperkenalkan iPhone 3G, yang memiliki fitur 3G sehingga kecepatan pengiriman data meningkat. Disamping itu pada tahun 2008, Apple juga meluncurkan iPhone SDK sehingga saat iPhone 3G diluncurkan, iPhone sudah mendukung aplikasi dari pihak ketiga. Pada tahun 2009, Apple mengenalkan iPhone 3Gs, yang memiliki tambahan kecepatan prosessor, dan memiliki kamera yang lebih baik sehingga mampu merekam video dan juga memiliki beberapa fitur tambahan lainnya. iPhone 4 diperkenalkan pada tahun 2010. Memiliki perubahan jauh dibanding iPhone sebelumnya dengan memiliki fitur retina display, dimana kepadatan pixel didalam layar iPhone 4 mencapai angka 312 dpi, padahal mata manusia hanya mampu menangkap kepadatan sampai angka 300 dpi saja. Disamping itu iPhone 4 juga memiliki kamera didepan, yang memungkinkan pengguna untuk melakukan video call.

1. **Fitur-fitur iPhone**

Pada subbab ini akan dibahas spesifikasi hardware yang mendeskripsikan iPhone secara umum dan fitur-fitur yang dimiliki oleh iPhone. Berikut adalah spesifikasi iPhone 4 secara teknis.

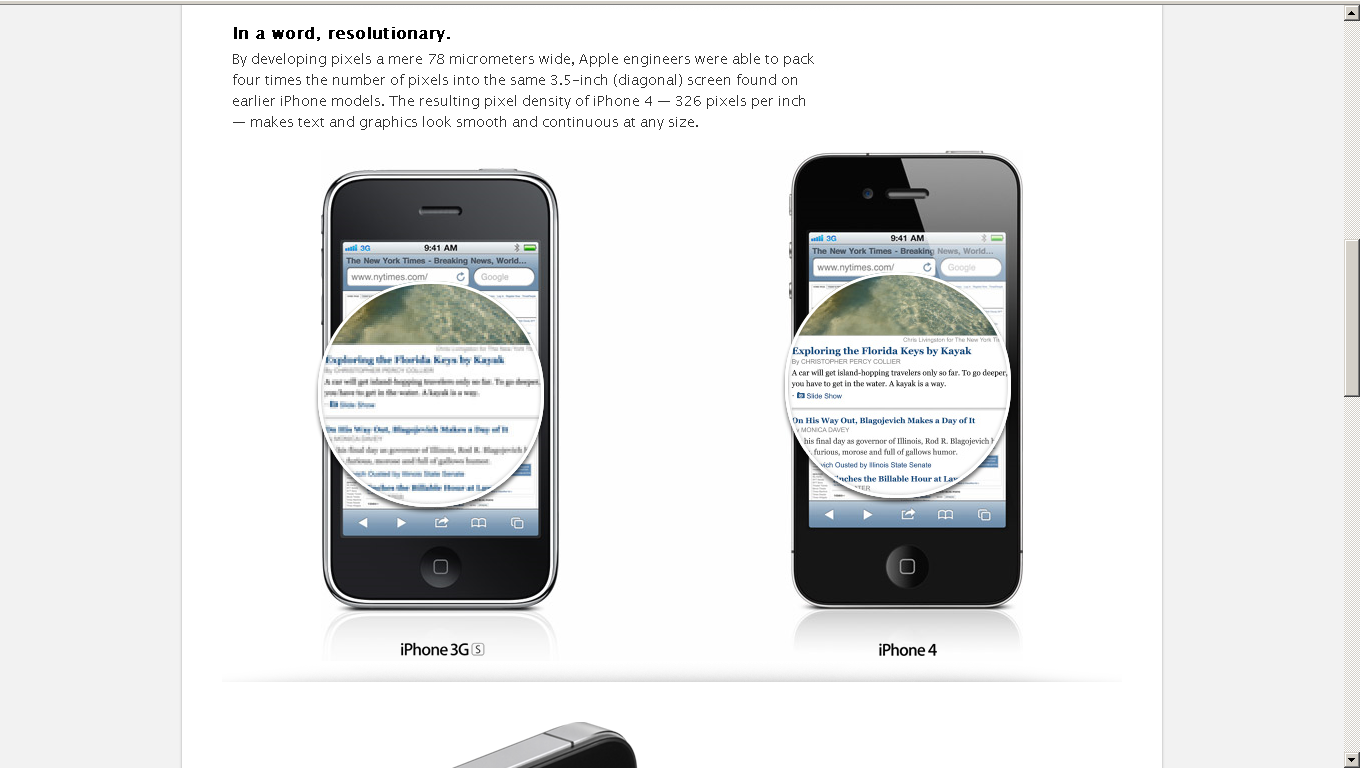
* Berat : 137 gram
* Dimensi : 115.2mm x 58.6mm x 9.3mm
* Layar : TFT capacitive touchscreen,

Retina display, 960 x 640 pixels

* RAM : 512MB
* OS : IPhonse OS (iOS)
* WLAN : Wi-Fi 802.11 b/g/n
* Koneksi Data : GPRS, EDGE, 3G, WLAN, Bluetooth, USB
* Kamera : 5MP
* Memori : 16GB / 32 GB

iPhone memilki touchscreen yang sudah menggunakan teknologi capacitive. Teknologi capacitive berbeda dengan teknologi sebelumnya, yaitu resistive. Pada resistive touchscreen, system akan mendeteksi sentuhan pengguna menggunakan semacam lapisan tipis yang akan menyentuh lapisan dibawahnya. Sentuhan antar kedua lapisan inilah yang akan dideteksi sebagai sentuhan dari pengguna. Sedangkan pada teknologi capacitive touchscreen, lapisan kaca dari layar akan dilapisi oleh konduktor, biasanya menggunakan Indium Tin Oxide (ITO). Karena tubuh manusia juga merupakan konduktor, maka saat menyentuh lapisan konduktor tersebut akan menimbulkan arus listrik. Arus inilah yang akan dideteksi sebagai sentuhan dari pengguna.

Salah satu fitur revolusioner dari iPhone 4 adalah fitur retina display. Fitur retina display berhasil memasukkan resolusi 960 x 640 pixels ke dalam layar iPhone berukuran 3,5 inci dengan cara menggunakan pixel sebesar 78 micrometer. Mata manusia hanya dapat melihat sebuah pixel hingga 300dpi, dengan resolusi yang diberikan oleh iPhone 4 sebesar 326dpi (lebih dari 300 pixel per inci) mengakibatkan Tampilan pada layar iPhone 4 semakin tajam.



**Gambar 2.2**

**Retina Display**

iPhone 4 juga menggunakan teknologi IPS (in-plane-switching), yaitu teknologi yang memberikan sudut pandang yang lebih besar pada LCD. Para pengguna iPhone dapat melihat layar iPhone dari berbagai sudut dan tetap mendapatkan tampilan yang tajam. Teknologi ini juga diterapkan pada tablet komputer milik Apple,yaitu pada iPad 2. Detail mengenai spesifikasi dan kemampuan dari iPad dan iPad 2 akan dijelaskan pada subbab berikutnya beserta dengan fitur-fitur yang dimiliknya.

* 1. **iPad**

iPad adalah sebuah komputer tablet yang diproduksi oleh Apple, Inc. Produk ini dirancang sebagai sebuah perangkat digital yang berada diantara smart phone dan laptop. Mesikpun menggunakan OS yang sama dengan iPhone tetapi iPad memiliki kelebihan dan kekurangan tersendiri dibandingkan dengan iPhone. Pada subbab ini akan dijelaskan mengenai perkembangan iPad dan fitur-fitur yang dimilikinya.

**2.2.1 Pengenalan iPad**

iPad adalah sebuah tablet yang dikenalkan pada publik pada 27 Januari 2010, dan pertama kali dijual pada publik pada 3 April 2010. Sampai saat ini sudah ada 2 generasi ipad yang diluncurkan oleh Apple, Inc. yaitu iPad 1 yang diluncurkan pada 2010 dan iPad 2 yang diluncurkan pada tahun 2011.



**Gambar 2.3**

**iPad**

Generasi iPad pertama sudah memiliki fitur yang cukup lengkap, dimulai dari fitur telepon, internet device, iPod, video Player, ebook reader dan sebagainya. Karena tidak memiliki sim card maka internet access hanya dapat diperoleh melalui wi-fi, hal ini telah diatasi oleh iPad generasi kedua yang telah dapat menggunkan sim card untuk mendapatkan internet access.

Sama seperti iPhone, iPad hanya memiliki 4 buah tombol, yaitu tombol home, volume, mute, dan sleep-wake. Hampir keseluruhan interaksi dengan pengguna dilakukan memalui touch screen. Berbeda dengan touch screen pada umumnya, iPhone dan iPad mendukung teknologi multi-touch yang memungkinkan iPhone dan iPad untuk mendeteksi dua atau lebih sentuhan pada saat yang bersamaan.

**2.2.2 Fitur-Fitur iPad**

Pada subbab ini akan dibahas spesifikasi hardware yang mendeskrpisikan iPad generasi Pertama dan generasi kedua serta fitur-fitur umum yang dimiliki oleh iPad. Tabel 2.1 menunjukkan spesifikasi iPad secara teknis.

**Tabel 2.1**

**Spesifikasi iPad 1 & iPad2**

|  |  |  |
| --- | --- | --- |
|  | iPad 1 | iPad 2 |
| Berat | 601 gram | 613 gram |
| Dimensi | 241,2 x 185,7 x 8.8 mm | 241,2 x 185,7 x 8.8 mm |
| Layar | 1024 x 768 pixels | 1024 x 768 pixels |
| Processor | A4 processor | A5 processor |
| Ram | 256MB | 512MB |
| OS | iOS | iOS |
| WLAN | Wi-Fi 802.11 a/b/g/n | Wi-Fi 802.11 a/b/g/n |
| Koneksi Data | Bluetooth, USB | Bluetooth, USB, 3G |
| Kamera | Tidak ada | 5x Digital Zoom |
| Memori | 16GB / 32GB / 64GB | 16GB / 32GB / 64GB |

Generasi iPad pertama memiliki beberapa kekurangan dibandingkan dengan iPad dari generasi kedua. Dapat dikatakan bahwa iPad 2, lebih cepat dan memiliki lebih banyak fitur dibandingkan dengan iPad 1. Salah satu perubahan besar dari iPad 2 adalah kekuatan dari processor dan ramnya. iPad 2 menggunakan 1 GHz dual core A5 processor dan ram sebesar 512 MB sedangkan iPad 1 menggunakan 1 GHz A4 processor dan ram sebesar 256 MB. Processor A5 memiliki clock speed yang dua kali lebih cepat dibandingkan dengan A4 dan 9 kali lebih baik ketika memproses grafik serta penggunaan baterei yang lebih hemat.

Perbedaan utama antara fitur iPhone dan iPad adalah pada Interface, fitur telepon dan resolusi layar. iPad memiliki beberapa interface yang tidak dimiliki oleh iPhone (sliding menu), iPad tidak dapat digunakan untuk menelpon walaupun pada iPad generasi kedua dapat menggunakan sim card. Resolusi layar yang dimiliki iPad adalah 1024 x 768 pixels, lebih besar dari pada iPhone.

* 1. **Xcode**

Xcode adalah IDE (Integrated Development Environment) utama untuk Mac. Xcode tidak hanya digunakan untuk mengembangkan aplikasi iPhone tetapi juga aplikasi untuk mac dan iPad. Didalam Xcode terdapat berbagai macam tool yang memiliki kegunaan berbeda-beda. iPhone SDK adalah Software Development Kit yang dibuat oleh Apple untuk membuat aplikasi iPhone. iPhone SDK diperkenalkan oleh Apple ke publik pada tahun 2008. Agar memudahkan pengguna Mac, maka iPhone SDK ditanamkan secara langsung ke dalam Xcode. Sejak diperkenalkan ke publik, iPhone SDK langsung mendapat tanggapan hangat dari para pengembang software. Setahun setelah diluncurkan, iPhone SDK telah didownload sebanyak lebih dari 1.000.000 kali. iPhone SDK kini telah dapat digunakan untuk membuat aplikasi untuk iPad dan iTouch yang juga menggunakan iOS.

iPhone SDK yang termasuk ke dalam Xcode terdiri atas beberapa aplikasi, yaitu: Xcode sebagai tempat pemrograman utama, Interface Builder untuk mendesain tampilan aplikasi yang dibuat, Instrument untuk menganalisa performa dari aplikasi yang dibuat, dan iPhone Simulator untuk mencoba aplikasi yang dibuat. Pada subbab ini akan dibahas mengenai Objective-C, yaitu bahasa yang digunakan untuk membuat aplikasi di iPhone, dan macam-macam aplikasi yang ada di Xcode.

* + 1. **Objective-C**

Objective-C adalah bahasa pemrograman yang merupakan pengembangan dari bahasa pemrograman C. Pertama kali diciptakan pada tahun 1980 dan digunakan oleh perusahaan komputer yang dikembangkan oleh Steve Jobs yaitu NeXT. Beberapa tahun kemudian Apple mengakuisisi NeXT dan menggunakan Objective-C sebagai bahasa pemrograman utama di Macintosh. Pada tahun 2008 Objective-C juga digunakan sebagai bahasa pemrograman untuk iPhone.

Karena Objective-C merupakan pengembangan dari bahasa pemrograman C maka syntax-syntax yang digunakan relatif sama. Yang membedakan keduanya adalah sifat Objective-C, yaitu Object Oriented. Pada Objective-C sebuah class terbagi menjadi 2 buah file, yaitu file dengan ekstensi \*.m dan \*.h. File dengan ekstensi \*.h dikenal dengan file Interface yang berisi deklarasi dari property dan prosedur yang bersifat public. Segmen program 2.1 menunjukkan struktur interface dari Objective-C.

**Segmen Program 2.1 Struktur Interface Objective-C**

1: #import “otherClassName.h”

2: #import <Foundation/Foundation.h>

3: @interface className : superClassName {

4: //variables declaration

5: }

6: +methodName;

7: +(returnType) methodName;

8: -(returnType) methodName:(paramType)paramName;

9: -(returnType) methodName:(param1Type)param1Name

param2Explain:(param2Type)param2Name;

10:@end

Struktur file interface pada Objective-C dapat dikelompokkan menjadi 3 bagian yaitu : bagian import, bagian deklarasi interface, dan bagian deklarasi method. Pada baris 1 dan 2 pada segmen program 2.1 merupakan contoh penggunaan import pada Objective-C. Terdapat 2 jenis import yang dapat digunakan yaitu import dengan menggunakan tanda “ ” dan import dengan menggunakan tanda < >. Import dengan tanda < > digunakan apabila class yang akan diimport bukan merupakan class standar milik Xcode, contoh dari class standar milik Xcode adalah Foundation.h dan UIKit.h. Bila ingin menginmpor class yang bukan merupakan standar milik Xcode maka digunakan tanda “ “, misalnya #import “WordNet.h” digunakan untuk mengimport class WordNet.

Baris 3-5 pada segmen program 2.1 diatas merupakan bagian deklarasi interface. Pada bagian ini terdapat deklarasi superClassName, pada Objectice-C yang bersifat Object Oriented semua class harus memiliki super class. Bila class tersebut bukan merupakan turunan dari class manapun maka ia merupakan turunan dari class NSObject. Baris 6-9 merupakan bagian deklarasi method yang dimiliki oleh interface. Terdapat 2 jenis deklarasi method pada Objective-C yaitu deklarasi dengan diawali simbol + dan deklarasi dengan diawail simbol -. Apabila sebuah method dideklarasikan dengan diawali simbol + maka method itu adalah sebuah method yang bersifat statik, dapat dipanggil tanpa harus membuat instance dari class tersebut. Tetapi apabila sebuah method dideklarasikan dengan diawali tanda – maka instance dari class harus dibuat terlebih dahulu sebelum dapat memanggil method tersebut. Baris 10 merupakan penutup dari class interface.

Apabila file \*.h berisi header dari method-method serta variable yang dimliki oleh sebuah class, maka file \*.m atau yang dikenal dengan file implementation ini merupakan tempat penulisan keseluruhan program. Line of code yang berisi perintah yang akan dieksekusi oleh sebuah method akan pada file implementation ini. Segmen program 2.2 menunjukkan struktur dari file implementation.

**Segmen Program 2.2 Struktur Implementation Objective-C**

1: #import “headerFileName.h”

2: @implementation className

3: +methodName{

4: //implementation

5: }

6: -(returnType) methodName:(paramType)paramName {

7: }

8: @end;

File implementation selalu diawali dengan baris 2 dan baris 8 segmen program 2.2, yaitu pendeklarasian implementation dan penutup dari class implementation. Baris 1 dari segmen program 2.2 merupakan line of code yang akan ditambahkan secara otomatis oleh Xcode. Setiap file implementation dari sebuah class harus mengimpor file header dari class tersebut. Semua class yang diimport pada file header akan dapat digunakan oleh file implementation, selain itu penambahan impor class juga dapat dilakukan secara langsung pada file implementation. Pada baris 3-5 dan 6-8 pada segmen program 2.2 menunjukkan contoh override dari method yang telah dideklarasikan pada file header. Method-method yang telah dideklarasikan pada file header harus dioverride pada file implementation. Method yang bersifat statik dan yang tidak bersifat statik memiliki cara override yang sama. Pada Objective-C juga dikenal istilah accessor dan mutator untuk mendapatkan dan mengubah nilai dari suatu variabel. Tidak seperti bahasa pemrograman berbasis object yang lain dimana accessor dan mutator harus dibuat secara manual, Obective-C memberikan kemudahan dengan membuatkan accessor dan mutator dari suatu variable secara otomatis dengan menambahkan sedikit line of code. Segmen program 2.3 menunjukkan contoh deklarasi accessor dan mutator untuk sebuah variable.

**Segmen Program 2.3 Deklarasi Accessor dan Mutator Objective-C**

1: @property(nonatomic,retain) variabelType variabelName;

2: @synthesize variabelName

Dengan menambahkan kedua line of code pada segmen program 2.3 maka Objective-C akan secara otomatis membuatkan accessor dan mutator untuk suatu variabel. Kedua baris line of code tersebut berada di dua buah file yang berbeda, baris pertama diletakkan pada file header di bagian deklarasi method sedangkan baris kedua diletakkan pada file implementation setelah deklarasi implementasi dilakukan. Nonatomic pada deklarasi property menandakan bahwa property tidak akan diakses secara thread-safe artinya terdapat kemungkinan akan terjadinya race-condition apabila dilakukan proses multi-threading. Sedangkan deklarasi retain menandakan bahwa property tersebut tidak akan dihapus dari memory oleh garbage collector. Setelah kedua line of code tersebut dituliskan maka accessor dan mutator dari variable tersebut dapat diakses secara langsung. Terdapat dua cara untuk mengakses accessor dan mutator tersebut. Kedua cara tersebut adalah :

**Segmen Program 2.4 Menggunakan Accessor dan Mutator Objective-C**

1: NSString \*x = [className variableName];

2: x = className.variableName;

3: [className setVariableName:x];

4: className.variableName = x;

Pada baris 1 segmen program 2.4 merupakan cara menggunakan accessor yang digunakan secara khusus oleh Objective-C, dimana variable dan class yang diakses menggunakan kurung siku. Sedangkan baris 2 pada segmen program 2.4 merupakan cara mengakses variable yang banyak dijumpai pada bahasa pemrograman yang lain, cara ini juga dapat digunakan pada Objective-C.

Pada Baris 3 segmen program 2.4 merupakan cara menggunakan mutator yang telah dibuatkan secara otomatis oleh Objective-C. Mutator ini memiliki nama method yang diawali oleh ‘set’ dan diikuti nama variable yang bersifat propercase. Baris 4 segmen program 2.4 merupakan cara kedua untuk menggunakan mutator pada Objective-C yaitu dengan mengakses nama variabel secara langsung, cara ini juga banyak dijumpai pada bahasa pemrograman yang lain.

Memory management merupakan hal yang sangat penting dalam pemrograman di iPhone, hal ini disebabkan karena memory iPhone yang sangat terbatas sehingga harus dimanfaatkan dengan sebaik mungkin. Objective-C memiliki garbage collector yang akan merelease variable dari memory apabila variabel tersebut sudah tidak lagi digunakan. Hal ini akan menjadi masalah apabila variabel tersebut telah direlease dan diakses oleh aplikasi, untuk mengatasi hal ini maka terdapat perintah retain.

**Segmen Program 2.5 Retain dan Release Objective-C**

1: NSString \*x = [[NSString alloc]init];

2: [x retain];

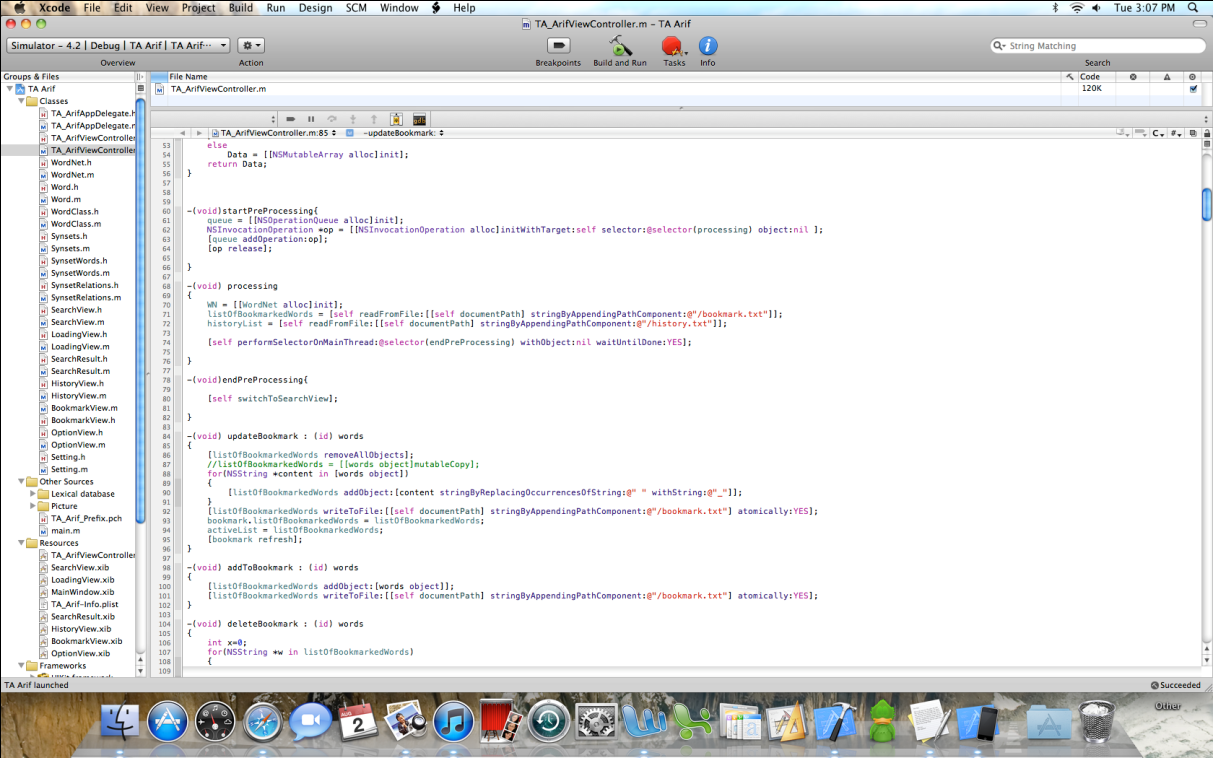
3: [x release];

Pada baris 1 segmen program 2.5 ditunjukkan bahwa variabel x mendapatkan inisialisasi dari class NSString, hal ini akan menyebabkan Objective-C melakukan reservasi memory untuk variabel x. Agar variabel x tidak terpengaruh oleh garbage collector maka ditambahkan perintah retain seperti pada baris 2. Dengan adanya perintah retain ini maka saat pembersihan memory dilakukan variabel x tidak akan dihapus walaupun tidak digunakan. Ketika aplikasi sudah tidak lagi menggunakan variabel x maka variabel x dapat dibuang dari memory dengan menggunakan perintah release seperti pada baris 3. Variabel yang tidak digunakan tetapi tetap berada dalam memory disebut dengan memory leak. Ketika mencapai batas tertentu maka aplikasi dapat berhenti secara tiba-tiba tanpa adanya peringatan apapun. Untuk mengatasi hal ini penggunaan tool analyzer akan sangat membantu serta penggunaan perintah retain dan release secara berkala.

Kelebihan lain dari Objective-C adalah adanya fasilitas delegate. Fasilitas delegate memungkinkan class untuk menyerahkan method yang seharusnya ia jalankan sendiri, untuk dijalankan oleh class lain yang menjadi delegate. Fungsi ini banyak digunakan dalam pemrograman di iPhone, antara lain untuk mendapatkan lokasi sentuhan dilayar untuk pengguna.

* + 1. **Penulisan Program di Xcode**

Xcode merupakan sebuah IDE yang dapat digunakan untuk membuat aplikasi pada MacOS dan iOS. Untuk membuat aplikasi pada kedua OS tersebut maka bahasa pemrograman yang harus digunakan adalah Objective-C tetapi bukan berarti Xcode tidak mendukung bahasa pemrograman lain. Xcode juga mendukung penggunaan bahasa pemrograman lain seperti C,C++,Java dan beberapa bahasa lainnya. Pada perkembangannya Xcode telah mencapai versi 4.0 dan masih dikembangkan hingga saat ini. Penulisan program pada Xcode memberi banyak kemudahan, diantaranya adapah pemberian warna yang berbeda pada setiap kata yang memiliki makna yang berbeda, sebuah nama variabel akan mendapat warna hijau sedangkan parameter akan berwarna pink. Selain itu pengetikkan suatu prosedur akan mendapat fasilitas auto complete dari Xcode, dengan menekan tambol tab maka prosedur tersebut akan dengan sendirinya dituliskan secara lengkap oleh Xcode beserta dengan jenis parameter yang diperlukan oleh prosedur tersebut. Gambar 2.4 merupakan contoh tampilan dari Xcode.



**Gambar 2.4**

**Tampilan Xcode**

Dapat dilihat pada gambar 2.4 bahwa Xcode akan secara otomatis mengelompokkan file-file dengan tipe tertentu. File header dan implementation akan dikelompokkan ke dalam folder classes. Apabila programmer ingin mengelompokkan lagi file header dan implementation yang ada maka programmer juga dapat membuat folder lagi di dalam folder classes tersebut. File tampilan dan media akan dikelompokkan ke dalam folder other resources.

Xcode adalah sebuah aplikasi yang ditujukan hanya untuk menuliskan program bagi aplikasi Mac dan iOS, untuk mendesain tampilan maka digunakan aplikasi lain yang bernama interface builder. Interface builder dan Xcode terhubung melalui sebuah file dengan ekstensi \*.xib yang dikenal dengan istilah file interface.

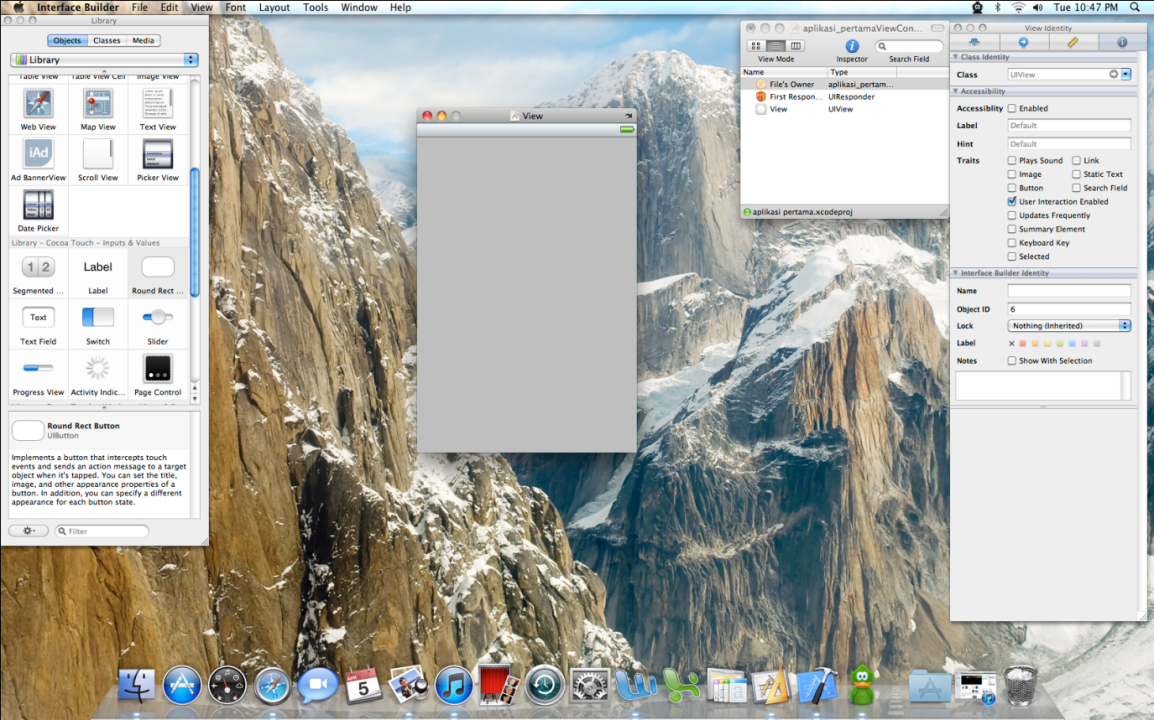
Untuk menjalankan aplikasi yang dibuat dapat dilakukan dengan memilih build and run atau dengan menekan command + return. Aplikasi yang dibuat dapat dijalankan pada simulator ataupun pada actual device seperti iPhone dan iPad. Apabila dijalankan memalui iPhone simulator maka dapat dilakukan debugging dengan mengaktifkan fitur breakpoints, fitur ini akan memberikan message apa bila terjadi error pada aplikasi yang sedang dijalankan. Sedangkan bila dijalankan melalui actual device maka fitur debugging atau yang lebih dikenal dengan istilah live debugging hanya bisa dilakukan apabila programmer terdaftar sebagai pengembang resmi di Apple dengan membayar $99 per tahun.

Selain fitur breakpoints Xcode juga menyediakan berbagai fitur lain seperti auto completion dan fitur analyzer. Fitur auto completion akan secara otomatis memberikan suggestion mengenai command yang akan ditulis dan melengkapi command tersebut apabila dipilih. Sedangkan fitur analyzer akan melakukan scanning pada line of code yang ada dan menunjukkan bagian-bagian yang memilki resiko terjadinya memory leak.

* + 1. **Interface Builder**

Selain Xcode aplikasi lain yang sangat membantu dalam pembuatan program pada Mac dan iOS adalah Interface Builder. Interface Builder digunakan untuk mendesain tampilan dari aplikasi yang dibuat, aplikasi tersebut bisa ditujukan untuk Mac, iPhone / iTouch, maupun iPad. Interface builder akan menyediakan ukuran frame yang sesuai dengan target device yang dipilih.

Pada Gambar 2.5 ditunjukkan bahwa terdapat beberapa window pada interface builder. Window yang berada disebelah kiri adalah window library yang berisi komponen-komponen yang telah disediakan oleh interface builder. Untuk memasukkan komponen-komponen tersebut ke dalam view aplikasi cukup dengan melakukan drag and drop dari library ke dalam view aplikasi. Window yang berada ditengah layar adalah view dari file \*.xib yang sedang aktif, sebuah file \*.xib memiliki 1 buah view. Setelah Window view terdapat window controller yang mengandung File’s Owner dan First Responder. Sedangkan window yang berada di paling kanan adalah window properties yang berisi properti-properti dari suatu komponen.



**Gambar 2.5**

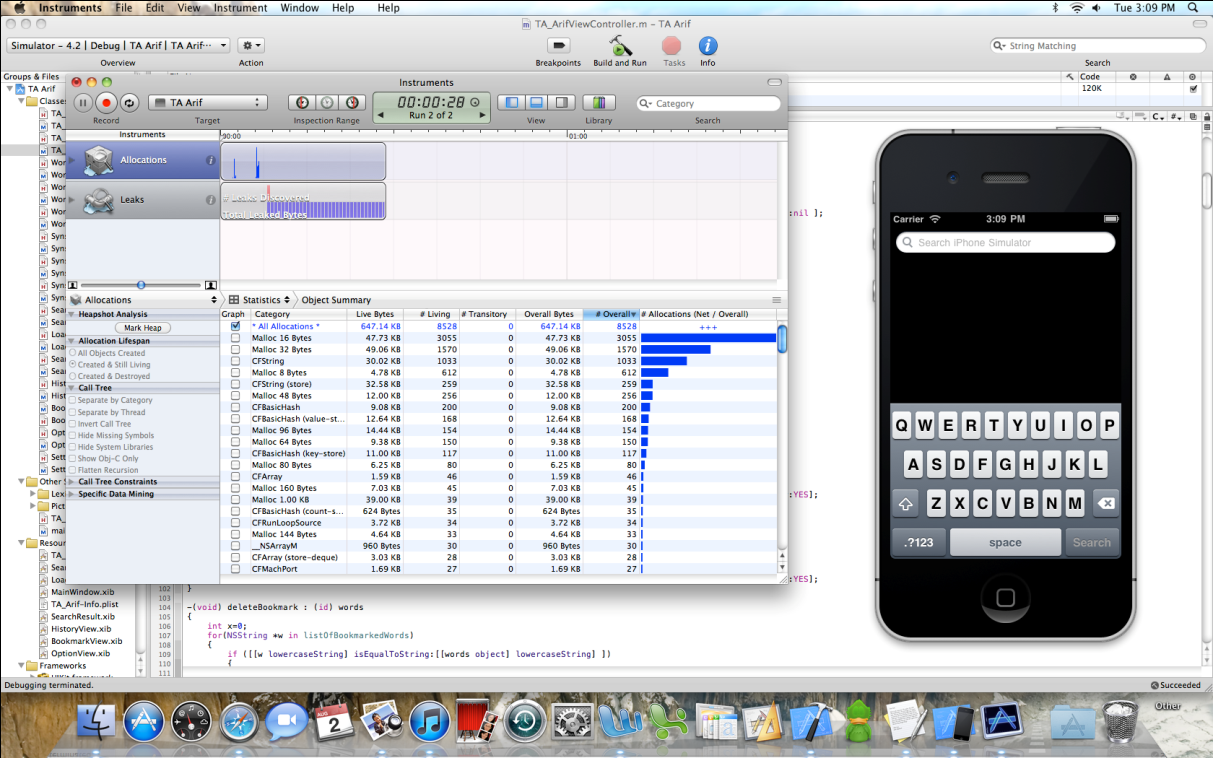
**Tampilan Interface Builder**

Suatu komponen dalam Interface Builder dapat dihubungkan dengan variabel bertipe IBOutlet yang dideklarasikan pada Xcode. Setiap komponen tersebut juga dapat diberi event yang didapatkan dari sebuah method yang dideklarasikan pada Xcode dengan tipe IBAction. Cara menghubungkan suatu komponen dengan Xcode adalah menarik garis dari komponen tersebut dengan controller yang bersangkutan. Penarikan garis dapat dilakukan dengan menahan tombol ctrl atau dengan menahan tombol mouse kanan dan melakukan drag dari komponen ke controller. Apabila garis ditarik dari controller File’s Owner dan dihubungkan dengan sebuah komponen maka akan muncul daftar IBOutlet yang dapat dihubungkan dengan komponen tersebut, tetapi jika garis ditarik dari komponen dan dihubungkan ke controller File’s Owner maka akan muncul daftar method yang bisa dilakukan. Sebuah komponen bisa memiliki lebih dari satu event, misalnya touch up inside, touch up outside,touch down inside,dl. Apabila sebuah action diberikan kepada komponen dengan cara menarik garis maka action tersebut akan masuk ke dalam event utama dari komponen tersebut.

Interface Builder juga menyediakan fasilitas untuk mendesain dalam portrait maupun landscape. Hal ini akan sangat berguna bagi aplikasi yang mengijinkan terjadinya rotasi layar, apabila terjadi rotasi maka tidak perlu membuat view yang berbeda tetapi cukup dengan merubah posisi dari komponan-komponen yang ada pada view tersebut dan mengubah ukuran agar sesuai dengan ukuran layar yang baru.

* + 1. **Instruments**

Salah satu tool yang disediakan oleh Xcode adalah tool Instruments, tool ini berguna untuk menganalisa kinerja dari aplikasi yang dibuat. Aplikasi ini dibuat berdasarkan DTrace tracing framework dari OpenSolaria dan sudah terintegrasi ke dalam Xcode. Gambar 2.6 menunjukkan tampilan Instruments.



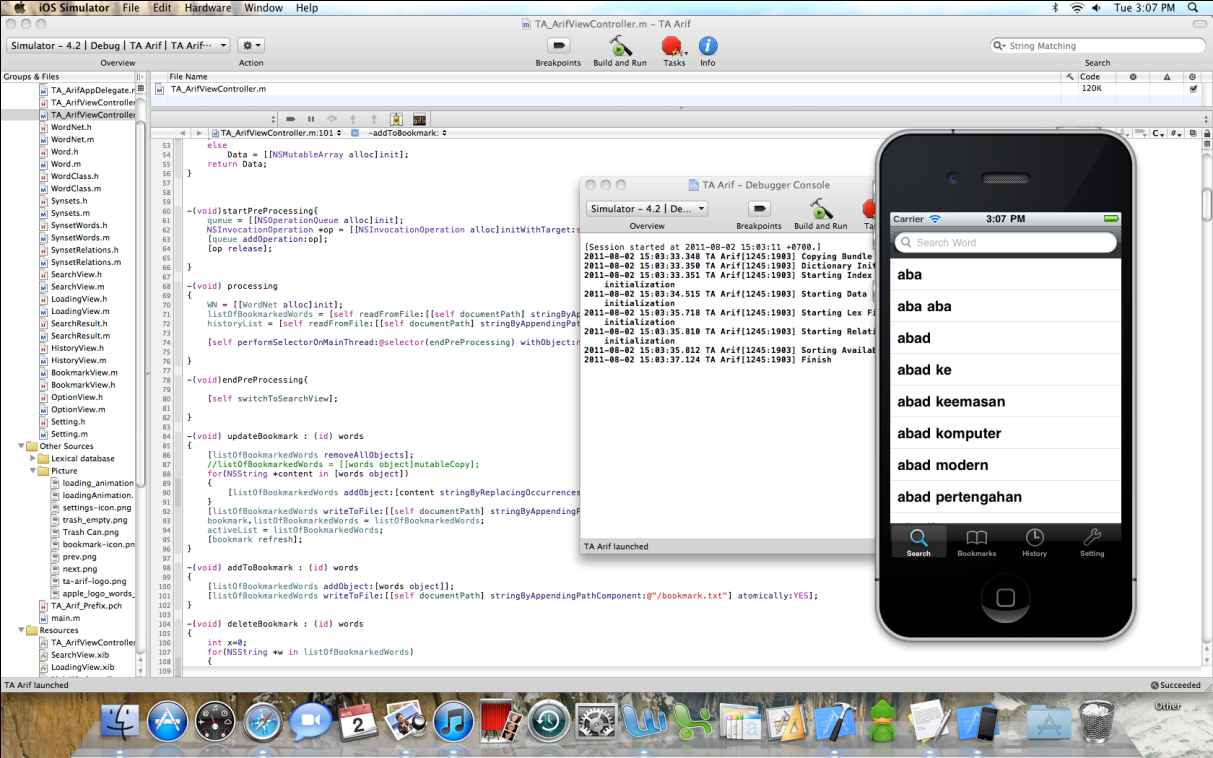
**Gambar 2.6**

**Tampilan Instruments**

Pada gambar 2.6 ditunjukkan bahwa instruments menampilkan timeline dari program yang disertai dengan grafik leaks. Grafik ini menunjukkan adanya memory leak pada aplikasi yang dibuat, semakin banyak leaks yang terjadi maka semakin tinggi grafik yang dihasilkan. Pada bagian bawah instruments akan ditunjukkan jumlah memory yang digunakan oleh class-class yang ada serta penggunaan class yang menyebabkan terjadinya memory leak, sayangnya tidak ditunjukkan line of code yang menyebabkan terjadinya memory leak. Instruments sangat berguna untuk meningkatkan kualitas dari aplikasi yang dibuat agar lebih cepat dan efisien. Dalam pengembangan aplikasi penggunaan instruments sangat disarankan.

* + 1. **iPhone Simulator**

iPhone Simulator merupakan bagian dari iPhone SDK yang terdapat didalam Xcode. Dengan adanya iPhone simulator maka programmer tidak perlu memiliki actual device untuk mencoba aplikasinya. Hampir semua fitur dari iPhone dapat dijalankan oleh simulator, misalnya penanganan touch dan multi touch pada layar, rotasi layar, akses internet dan berbagai fitur yang lain. Gambar 2.7 menunjukkan tampilan dari iPhone simulator.

** Gambar 2.7**

**iPhone Simulator**

Beberapa fitur yang tidak dapat dilakukan oleh simulator adalah kamera, accelerometer, GPS dan beberapa fasilitas lain yang membutuhkan hardware yang tidak terdapat pada mac. Sehingga untuk uji coba fitur-fitur tersebut harus menggunakan iPhone dan memanfaatkan fitur live debugging. Tetapi dengan adanya simulator ini maka debug untuk fitur-fitur yang lain dapat dilakukan dengan memanfaatkan fitur breakpoints milik Xcode. Fitur breakpoint dapat melakukan trace terhadap jalannya program.

* + 1. **Arsitektur iPhone SDK**

API pada iPhone memiliki banyak kesamaan dengan API milik Mac, keduanya terbagi menjadi 4 bagian. Yang membedakan kedua API tersebut adalah API lapisan teratas, apabila pada Mac lapisan teratasnya adalah Cocoa maka pada iPhone lapisan teratasnya adalah Cocoa Touch. Perbedaan ini disebabkan karena adanya perbedaan input pada kedua device, pada Mac input berupa pengetikkan pada keyboard sedangkan pada iPhone inputan berupa sentuhan pada layar device. Berikut ini adalah susunan arsitektur iPhone SDK:

* Cocoa Touch

Cocoa Touch berisi fasilitas-fasilitas untuk berinteraksi dengan pengguna, seperti menerima inputan dan mengatur tampilan. Lapisan teratas dari arsitektur iPhone ini berisi: Multi-Touch events and controls, Accelerometer support, View hierarchy, Localization, dan Camera Support.

* Media

Lapisan Media seperti namanya berisi beberapa kumpulan API dan framework untuk memutar dan menampilkan video, gambar dan suara. Isi dari lapisan Media adalah: Open AL, audio mixing and recording, Video Playback, Image file formats, Quartz, Core Animation, dan OpenGL ES.

* Core Services

Lapisan Core Services berisi beberapa fasilitas-fasilitas dasar, dan sudah mulai berhubungan dengan hardware. Fasilitas-fasilitas yang ada dalam Core Services adalah: Networking, Embedded SQLite database, Core Location, Threads, dan Core Motion.

* OS X Kernel

Pada dasarnya system operasi yang dijalankan di iPhone sama dengan yang dijalankan di Mac, yaitu OS X. Oleh karena itu lapisan paling bawah dari arsitektur iPhone SDK adalah OS X Kernel. Dalam lapisan ini, hampir semuanya berhubungan langsung dengan hardware. Isi dari lapisan ini adalah: TCP/IP, Sockets, Power Management, File System, dan Security.

* 1. **WordNet**

WordNet[[1]](#footnote-2) adalah sebuah kamus yang didesain menyerupai sebuah thesaurus. Pada WordNet sebuah kata disimpan dalam bentuk sinonim set (synset). Synset adalah kata-kata yang memiliki bentuk yang berbeda tetapi memiliki arti kata yang sama. Tetapi berbeda dengan thesaurus, WordNet tidak hanya mengandung kumpulan kata-kata yang memiliki arti yang sama tetapi juga mengandung informasi lain berupa relasi dari kata-kata tersebut beserta glossnya.

* + 1. **Sejarah Pembentukan WordNet**

Pada umumnya kamus dibuat dengan mengurutkan daftar kata yang ada secara alphabetical tanpa memperhatikan arti dari kata tersebut. Hal ini memberikan kemudahan bagi pembuat dan pengguna kamus. Tetapi sistem ini memiliki kekurangan yaitu pencarian kata secara alphabetical akan memakan waktu sehingga menyebabkan masyarakat malas mencari kata dalam suatu kamus.

Seiring dengan berkembangnya teknologi maka masalah pencarian kata yang memakan waktu telah dapat diatasi oleh komputer. Komputer dapat melakukan pencarian kata pada sebuah kamus elektronik dengan kecepatan yang sangat tinggi. Walaupun demikian sistem yang digunakan oleh kamus tersebut tetaplah sistem konvensional yang tidak memperhatikan arti dari kata tersebut. Para ahli bahasa akhirnya mengusulkan suatu sistem baru yang disebut WordNet dimana sebuah kamus dikombinasikan dengan informasi lexical dan didukung oleh komputer berkecepatan tinggi. WordNet menyerupai sebuah kamus konvensional dengan memberikan informasi mengenai definisi besera contoh penggunaan kata yang ada pada synset. Tetapi kelebihan dari WordNet adalah kemampuannya menyajikan informasi lexical dengan memberikan relasi-relasi yang ada beserta sinonim yang dimiliki kata tersebut. Dapat dikatakan bahwa WordNet merupakan perkembangan dari kamus dan thesarurus dimana keduanya disatukan dan mendapat tambahan informasi berupa relasi kata.

* + 1. **Lexical Database**

Lexical Database yang dimiliki oleh WordNet dituliskan dalam format ASCII. Hal ini dilakukan untuk memudahkan pengaksesan dalam berbagai bahasa pemrograman. File Lexical Database ini dapat dilihat dengan menggunakan text editor pada berbagai OS, misalnya TextEdit pada Mac dan WordPad pada Windows. Kelemahan dari sistem penyimpanan dalam format ascii ini adalah file Lexical Database akan sangat mudah diubah oleh orang awam.WordNet bahasa Indonesia yang digunakan pada Tugas Akhir ini mengandung 9 buah file dengan ukuran 11,4 MB. Tabel 2.2 menunjukkan detail Lexical Database Files.

**Tabel 2.2**

**Daftar File Lexical Database**

|  |  |  |
| --- | --- | --- |
| Nama File | Fungsi | Size (KB) |
| index.noun | Index file untuk kata noun | 1,952 |
| index.verb | Index file untuk kata verb | 1,215 |
| index.adj | Index file untuk kata adj | 54 |
| index.adv | Index file untuk kata adv | 470 |
| data.noun | Data file untuk kata noun | 5,352 |
| data.verb | Data file untuk kata verb | 1,992 |
| data.adj | Data file untuk kata adj | 646 |
| data.adv | Data file untuk kata adv | 61 |
| lexnames | Mencatat lexnames | 1 |

Semua file Lexical Database yang ada pada tabel 2.2 ditulis dengan menggunakan format ASCII. Untuk pemisah setiap kata pada index file maupun synset pada data file digunakan karakter enter (newline). Sedangkan untuk pemisah antar field yang ada digunakan karakter spasi.

Pencarian sebuah kata akan melalui dua tahap proses yaitu: pencarian kata pada setiap index file dan ekstraksi synset data pada data file. Pencarian kata pada setiap index file dilakukan untuk mengatasi kemungkinan sebuah kata dapat berada pada lebih dari 1 class kata.

* + - 1. **Index File**

Index File berisi daftar semua kata pada WordNet sesuai dengan class kata masing-masing (noun,verb,adj,adv). Kata-kata tersebut disusun urut sesuai dengan alphabet dan ditulis dengan huruf kecil. Setiap kata yang ada dipisahkan oleh karakter enter (newline). Index File diawali dengan keterangan mengenai versi WordNet dan license agreement mengenai penggunaan WordNet tersebut, license agreement ini berjumlah 29 baris. Setelah license agreement tersebut selesai maka dimulailah baris-baris yang berisi data dari WordNet, format dari baris-baris tersebut adalah sebagai berikut:

lemma class synset\_cnt p-cnt [ptr\_symbol...] sense\_cnt

tagsense\_cnt synset\_offset [synset\_offset...]

Berikut ini adalah penjelasan dari format di file index:

* Lemma

Kata yang ditulis dalam huruf kecil. Apabila kata ini terpisah oleh spasi maka spasi tersebut akan diganti dengan garis bawah ( \_ ), contohnya abad\_ke, abah\_abah dan sebagainya.

* Class

Kategori dari bentuk kata. Terdapat 4 kemungkinan untuk class ini yaitu: karakter “n” untuk class noun, karakter “v” untuk class verb, karakter “a” untuk class adjective dan class “r” untuk class adverb.

* Synset\_cnt

Merupakan jumlah dari synset yang dimiliki oleh lemma. Juga merupakan jumlah sense yang dimiliki.

* P\_cnt

Jumlah pointer yang dimiliki oleh sebuah lemma. Pointer yang dimaksud adalah pointer yang menunjukkan relasi-relasi yang dimiliki.

* Ptr\_symbol

Simbol-simbol pointer yang dimiliki oleh lemma. Setiap symbol pointer memiliki arti tersendiri, hal ini akan dijelaskan lebih detail pada subbab 2.4.2.3. Apabila lemma tidak memiliki relasi apapun maka p\_cnt akan bernilai 0 dan field ini akan dihilangkan.

* Sense\_ cnt

Jumlah dari sense yang dimiliki oleh lemma.

* Tagsense\_cnt

Jumlah penggunaan lemma yang disorting berdasarkan frekuensi penggunaan pada teks *semantic concordance*.

* Synset\_offset

Byte offset dari synset yang dimiliki oleh lemma. Setiap synset\_offset terhubung dengan synset yang berbeda data file. Synset offset terdiri dari 8 digit bilangan integer.

Berikut ini adalah contoh kata pada index file disertai dengan penjelasaan mengenai field-field yang dimiliki

abc n 3 2 #p %m 3 0 05321050 04785693 01743487

Field yang dimiliki:

* Lemma : abc
* Class : n (noun)
* Synset\_cnt : 3
* P\_cnt : 2
* Ptr\_symbol : #p dan %m

(sesuai dengan jumlah pada p\_cnt)

* Sense\_cnt : 3
* Tagsense\_cnt : 0
* Synset\_offset : 05321050, 04785693, 01743487

(Sesuai dengan jumlah pada synset\_cnt)

* + - 1. **Data File**

Data File berisi data-data yang dimiliki oleh sebuah synset. Dimulai dari sinonim, gloss, example dan relasi-relasi lexical yang dimiliki oleh synset tersebut. Pada awal setiap data file juga terkandung keterangan mengenai versi WordNet dan juga license agreement sejumlah 30 baris. Setelah itu barulah setiap baris mewakili satu buah synset. Format synset tersebut adalah sebagai berikut:

Synset\_offset lex\_filenum ss\_type w\_cnt word lex\_id

[word lex\_id...] p\_cnt [ptr...] (frames..) | gloss

Berikut ini adalah penjelasan dari format di file data:

* Synset\_offset

Byte Offset dari synset yang berupa 8 digit bilangan integer.

* Lex\_filenum

Dua digit integer yang mencatat nomor file lexnames yang mengandung synset ini. Lexname akan dijelaskan pada subbab 2.4.2.4.

* Ss\_type

Berupa 1 karakter yang menunjukkan class kata. Memiliki nilai yang sama dengan field class pada index file.

* W\_cnt

Berupa 2 digit hexadesimal yang menunjukkan jumlah kata pada synset.

* Word

Kata-kata yang berada didalam suatu synset dan ditulis dalam bentuk ASCII. Apabila kata tersebut terpisah oleh spasi maka akan digantikan dengan garis bawah (\_). Word bersifat case-sensitive berbeda dengan lama pada index file yang ditulis dengan huruf kecil.

* Lex\_id

Berupa 1 digit hexadesimal integer yang jika ditambahkan pada lemma, akan memberikan karakteristik unik pada arti kata dalam lexical file. Biasanya berawal dari 0 dan bertambah sesuai dengan jumlah munculnya word yang ditambahkan pada file yang sama.

* P\_cnt

Berupa 3 digit bilangan integer yang menunjukkan jumlah pointer yang dimiliki oleh synset. Pointer yang dimaksud adalah pointer relasi. Apabila synset tidak memilki pointer apapun maka akan bernilai 000.

* Ptr

Berfungsi menghubungkan suatu synset dengan synset lain dengan format sebagai berikut:

Ptr\_symbol synset\_offset class source/target

Format tersebut memiliki kesamaan dengan format pada index file. Ptr\_symbol menyatakan relasi yang dimiliki dengan synset\_offset, class menunjuk class tempat synset\_offset berada. Source/target berupa 4 digit bilangan hexadesimal, 2 digit pertama menunjukkan nomor word pada synset asal dan 2 digit terakhir menunjukkan nomor word pada synset target. Apabila Source/Target bernilai 0000 berarti kedua synset memiliki hubungan semantic. Tetapi apabila tidak bernilai 0000 maka kedua synset memiliki hubungan lexical. Nomor word berhubungan dengan field word pada synset, dari kiri ke kanan dimulai dari 1.

* Frames

Frames hanya terdapat pada data.verb saja. Frames memiliki format:

F\_cnt + f\_num w\_num [+ f\_num w\_num.. ]

F\_Cnt menunjukkan jumlah dari frame yang terdaftar, F\_cnt berupa 2 digit bilangan integer. F\_num merupakan 2 digit bilangan desimal menunjukkan nomor frame yang digunakan. W\_num berupa 2 digit bilangan hexadesimal yang menunjukkan word pada synset yang dijelaskan oleh field frame. Bila w\_num bernilai 00 maka f\_num mengacu untuk semua word pada synset.

* Gloss

Gloss merupakan definisi dari sebuah synset dan dapat disertai oleh contoh kalimat. Gloss diawali dengan menggunakan sumbol “|”.

Berikut ini dalah contoh data file dari data.noun:

05321050 03 n 0f abc 3 abece 0 abjad 2 aksara 2 alfabet 0 alif\_ba\_ta 0 alif\_bata 0 fonem 0 hiroglif 0 huruf 4 ideograf 0 kritogram 0 lambang\_bunyi 0 leter 2 piktograf 0 003 %m 03473223 n 0000 %m 05164323 n 0000 #p 05077790 n 0000 | hurufhuruf yang dikembangkan oleh bangsa Inggris yang didasarkan pada alfabet Romawi yang usianya kirakira 2.500 tahun

Data dari tiap field adalah sebagai berikut:

* Synset\_offset : 05321050
* Lex\_filenum : 03
* Ss\_type : n (noun)
* W\_cnt : 0f (15 desimal)
* Word : abc, abece, abjad, aksara, alphabet, alif\_ba\_ta,

alif\_bata, fonem, hiroglif, huruf, ideograf,

kritogram, lambang\_bunyi, leter, piktograf

(15 word sesuai dengan w\_cnt)

* Lex\_id : 3, 0, 2, 2, 0, 0, 0, 0, 0, 4, 0, 0, 0, 2, 0
* P\_cnt : 003
* Ptr
  + Ptr\_symbol : %m, %m, %p
  + Synset\_offset : 03473223, 05164323, 05077790
  + Class : n (noun), n (noun), n (noun)
  + Soucer/Target : 0000, 0000, 0000
* Frames : -
* Gloss : hurufhuruf yang dikembangkan oleh

bangsa Inggris yang didasarkan pada

alfabet Romawi yang usianya

kirakira 2.500 tahun

Berikut ini adalah contoh data file dari data.verb:

00001740 29 v 01 mengerpus 0 000 01 + 01 00 | menghukum dengan cara memasukkan prajurit ke dalam kerpus menghukum dalam kurungan

Data dari tiap field adalah sebagai berikut:

* Synset\_offset : 00001740
* Lex\_filenum : 29
* Ss\_type : v (verb)
* W\_cnt : 01
* Word : mengerpus

(1 word sesuai dengan w\_cnt)

* Lex\_id : 0
* P\_cnt : 000
* Ptr : -
* Frames : 01 + 01 00
  + F\_cnt : 01
  + F\_num : 01
  + W\_num : 00
* Gloss : menghukum dengan cara memasukkan prajurit

ke dalam kerpus menghukum dalam kurungan

* + - 1. **Pointer Symbol**

WordNet menggunakan simbol-simbol untuk melambangkan hubungan lexical yang ada. Setiap relasi memilki simbol yang berbeda-beda, Tabel 2.3 menunjukkan relasi yang dimiliki oleh WordNet.

**Tabel 2.3**

**Pointer Simbol pada WordNet**

|  |  |  |
| --- | --- | --- |
| Relasi | Pointer | Bentuk kata |
| Antonim | ! | Noun, Verb, Adjective |
| Hipernim | @ | Noun, Verb |
| Instance Hipernim | @i | Noun |
| Hiponim | ~ | Noun, Verb |
| Instance Hiponim | ~i | Noun |
| Member Holonim | #m | Noun |
| Substance Holonim | #s | Noun |
| Part Holonim | #p | Noun |

**Tabel 2.3**

**(Lanjutan)**

|  |  |  |
| --- | --- | --- |
| Relasi | Pointer | Bentuk kata |
| Member Meronim | %m | Noun |
| Substance Meronim | %s | Noun |
| Part Meronim | %p | Noun |
| Atribut | = | Noun, Adjective |
| Derivationally Related Form | + | Noun |
| Domain of Synset - Topic | ;c | Noun, Verb, Adjective |
| Member of this domain - Topic | -c | Noun |
| Domain of Synset - Region | ;r | Noun, Verb, Adjective |
| Member of this domain - Region | -r | Noun |
| Domain of Synset - Usage | ;u | Noun, Verb, Adjective |
| Member of this domain - Usage | -u | Noun |
| Entailment | \* | Verb |
| Cause | > | Verb |
| Also See | ^ | Verb, Adjective |
| Verb Group | $ | Verb |
| Similar To | & | Adjective |
| Participle of Verb | < | Adjective |
| Pertainim (Perains to Noun) | \ | Adjective |

Pada WordNet Bahasa Indonesia yang digunakan pada Tugas Akhir ini hanya mengandung antonim, hipernim, hiponim, holonim, dan meronim. Antonim adalah sebuah relasi yang dimiliki oleh sebuah kata yang merujuk pada kata yang memiliki arti yang berlawanan, contohnya kehidupan dan kematian. Hipernim adalah relasi yang merujuk pada kata yang sifat nya lebih umum sedangkan hiponim merupakan relasi yang merujuk pada kata yang sifatnya lebih spesifik. Contoh hipernim adalah : gurun pasir dan gurun sedangkan contoh hiponim adalah gurun dan gurun pasir, kedua relasi tersebut saling berhubungan satu sama lain.

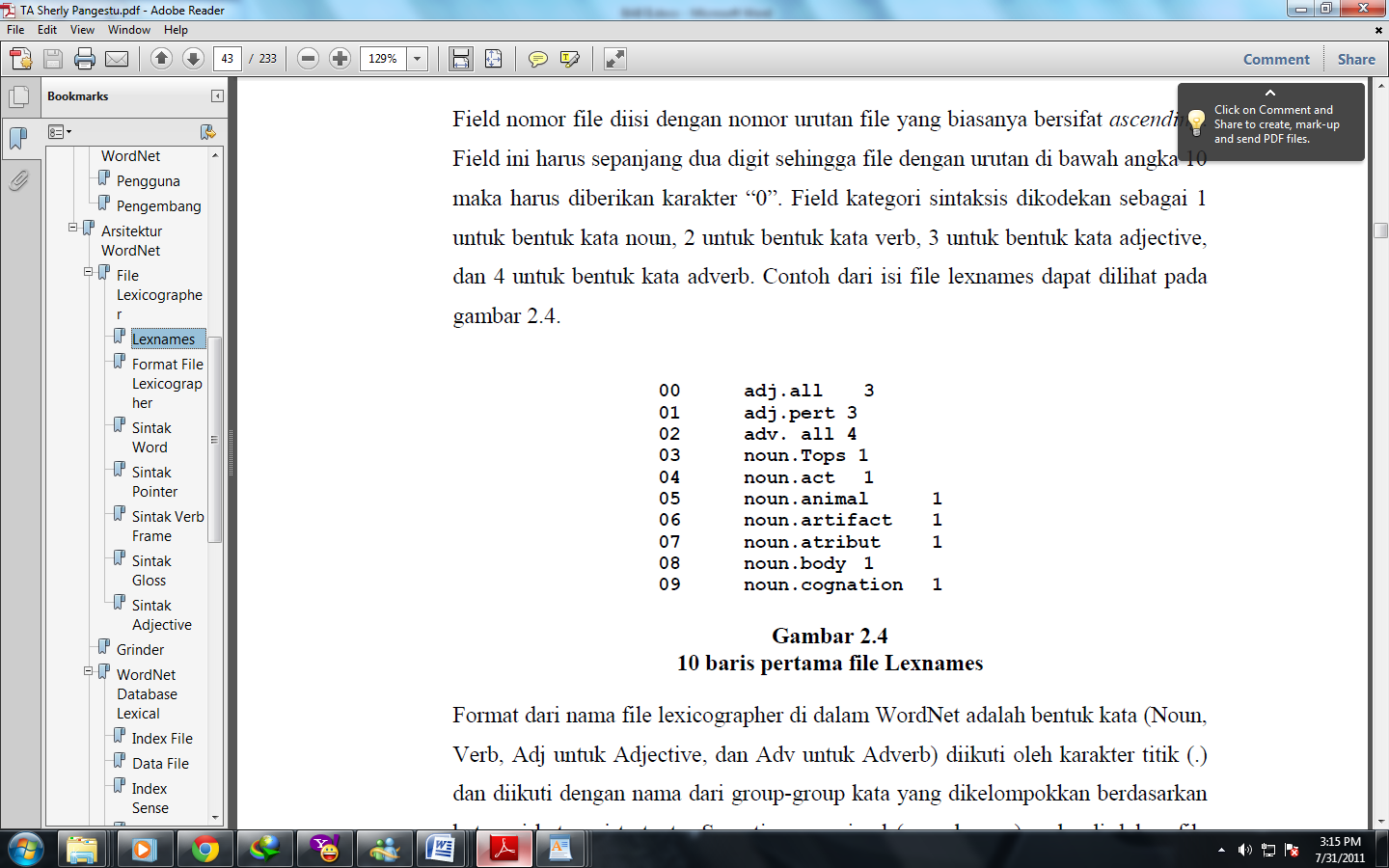
Holonim adalah sebuah sebuah relasi dimana sebuah kata merupakan bagian yang lebih besar yang disusun oleh relasi meronim sedangkan meronim adalah sebuah relasi dimana sebuah kata merupakan bagian dari sesuatu. Contoh holonim: mata dan retina, mata merupakan sesuatu yang tersusun oleh retina. Contoh dari meronim adalah kebalikan dari holonim yaitu retina dan mata, retina merupakan bagian dari mata. Holonim dan meronim juga merupakan relasi yang tidak dapat dipisahkan.Relasi-relasi tersebut saling berhubungan satu sama lain dan diwakilkan dengan menggunakan simbol. Setiap simbol bersifat unik dah hanya mewakilkan satu buah relasi saja.

* + - 1. **Lexnames**

Lexnames merupakan hasil proses grinder terhadap lexicographer file. Lexnames ini mengandung informasi class dari kata diikuti dengan kelompok kata. Format dari lexnames adalah :

File\_number Nama\_File Kategori\_Sintaksis

Field file\_number berupa 2 digit integer yang merupakan nomor urut file dan disusun secara ascending. Angka yang hanya 1 digit harus menambahkan angka 0 didepannya. Nama\_file merupakan field dengan tipe string yang memiliki struktur class kata diikuti dengan karakter “.” dan kategori kelompok yang dikelompokkan berdasarkan kategori tertentu. Gambar 2.8 menunjukkan isi dari file lexnames.



**Gambar 2.8**

**10 baris pertama dari lexnames**

Seperti pada gambar 2.8 nama\_file berupa tipe class yang diikuti dengan karakter “.” dan grup kata. Misalnya noun.animal berarti kata dengan class noun yang termasuk ke dalam class hewan. Field kategori\_sintaksis akan diisi sesuai dengan tipe class kata, bila noun maka field akan bernilai 1, bila verb maka field akan bernilai 2, bila adjective maka field akan bernilai 3, bila adverb maka field akan bernilai 4.

* 1. **Pembuatan WordNet Bahasa Indonesia**

WordNet Bahasa Indonesia merupakan hasil penelitian dari para mahasiswa STTS. Para mahasiswa tersebut melakukan pembentukan synset, mendapatkan definisi dan membuat aplikasi untuk melakukan perubahan pada WordNet. Terdapat empat orang mahasiswa yang melakukan pembuatan WordNet bahasa Indonesia, yaitu: Andy Saputra, Erick Pranata, Heru Raharjo, Theofilus Christyawan. Pembuatan synset dilakukan dengan memanfaatkan resource yang berasal dari KBBI (Kamus Besar Bahasa Indonesia), Tesaurus dan kamus bilingual bahasa Indonesia dan Inggris. Pembuatan synset tersebut ditangani oleh Andy Saputra dan dalam penelitiannya terdapat beberapa kendala yang dihadapi ketika synset dibentuk dengan menggunakan resource monolingual :

1. Data dalam Tesaurus tidak lengkap dan tidak konsisten sehingga membutuhkan proses lebih lanjut dan asumsi yang ditentukan sendiri. Inkonsistensi data dari Tesaurus ini dapat terlihat pada tercampurnya sinonim dan hiponim pada pasangan dalam Tesaurus. Hal ini harus ditangani dengan konsep hubungan sinonim yang digunakan. Ketidaklengkapan Tesaurus dapat terlihat pada beberapa kata yang tidak memiliki entry di dalam Tesaurus, namun disebutkan pada pasangan dari kata lain sehingga tidak bisa dilakukan proses pemeriksaan hubungan sinonimnya.
2. Tidak kongruennya data dari KBBI dan Tesaurus sehingga mempengaruhi proses penggabungan data dari KBBI dan Tesaurus. Contoh dari tidak kongruennya data dari resource yang digunakan adalah misalkan dalam KBBI, lema ‘abad’ memiliki 4 sense yang berbeda. Sedangkan dalam Tesaurus, lema ‘abad’ hanya memiliki 1 pasangan. Karena kurangnya informasi, maka tidak diketahui lema ‘abad’ yang dimaksud dalam Tesaurus ini merujuk pada sense yang mana dari lema ‘abad’ pada KBBI. Hal ini mengurangi akurasi dari proses penggabungan.
3. Proses ekstraksi resource yang kurang sempurna menyebabkan beberapa kata yang tidak dapat diproses dengan akurat. Misalkan saja, terdapat kata ‘panji-panji’, ‘panjipanji’ dan ‘panjipan-ji’ yang akan dianggap kata yang berbeda oleh program, sementara sebenarnya ketiga contoh tersebut merujuk pada satu kata yang sama.

Sedangkan pembentukan synset dengan memanfaatkan resource bilingual dengan menggunakan word sense disambiguation pun memiliki kendala tersendiri. Beberapa kendala tersebut antara lain adalah:

1. Keterbatasan kualitas dan ketersediaan kamus yang memadai, sehingga hasil terjemahan dari Bahasa Indonesia menjadi Bahasa Inggris dan sebaliknya kurang baik. Tidak semua terjemahan dari kamus yang digunakan tersusun dalam bentuk kata, namun terdapat juga terjemahan dalam bentuk definisi yang memerlukan proses parsing. Hal ini cukup rentan menimbulkan kesalahan penerjemahan, terutama untuk penerjemahan kata Bahasa Indonesia menjadi Bahasa Inggris.
2. Terdapat perbedaan kelas kata antara kata dalam Bahasa Indonesia (dalam KBBI) dan Bahasa Inggris (dalam kamus) sehingga ada beberapa kata yang tidak ditemukan terjemahannya (karena proses penerjemahan dilakukan hanya terhadap kata yang kelas katanya sama saja). Contohnya adalah kata ‘hamil’ yang memiliki kelas kata ‘verb’ di KBBI, sedangkan terjemahannya, ‘pregnant’ memiliki kelas kata ‘adjective’. Hal ini mengakibatkan ‘pregnant’ tidak dianggap terjemahan yang valid untuk ‘hamil’.

Untuk definisi suatu kata ditangani oleh Theofilus Christyawan. Definisi dari suatu kata didapatkan dari melakukan crawling / fetching web-web site berbahasa Indonesia yang dihasilkan oleh google. Kata yang ingin dicari akan dimasukkan ke dalam google search dan hasil web site yang didapatkan akan diambil sebagai sumber data. Hasil yang didapatkan meliputi dari huruf A hingga Z. Resource tersebut kemudian akan diproses untuk didapatkan definisi dari kata-kata yang ada dan dimasukkan ke dalam sense yang bersangkutan.

Erick Pranata menangani relasi hipernim dan hiponim, sedangkan Heru Raharjo menangani relasi holonim dan meronim. Keduanya menggunakan KBBI dan wikipedia sebagai sumber datanya. Hasil penelitian keempat orang tersebut menjadi dasar terbentuknya WordNet bahasa Indonesia. Penelitian tersebut kemudian diteruskan oleh Jessica Felani Wijoyo dengan membuat editor untuk WordNet.

* 1. **Levenshtein Distance**

Levenshtein Distance atau yang lebih dikenal dengan istilah edit distance adalah sebuah algoritma yang digunakan untuk menghitung perbedaan antara dua buah sequence. Pada Tugas Akhir ini levenshtein distance digunakan untuk menghitung biaya yang digunakan untuk merubah sebuah kata ke kata yang lain dengan cara menambahkan atau mengurangi huruf dan menukar posisi huruf yang ada pada sebuah kata. Hasil perhitungan dari levenshtein distance akan digunakan untuk menangani kemungkinan terhadinya kesalahan pengetikan oleh pengguna dan memberikan daftar kata yang mungkin ingin diketik oleh pengguna.

Setiap terjadi penambahan atau pengurangan huruf dan perubahan posisi huruf pada suatu kata maka levenshtein akan menambahkan biaya sebesar 1. Algoritma ini akan mencoba semua kemungkinan yang dapat dilakukan untuk merubah suatu kata menjadi kata yang lain dan mengambil biaya terkecil yang bisa didapatkan.

1. Pangestu. S., *Pembuatan Prototype Database Lexical untuk Bahasa Indonesia yang mengacu pada Wordnet*. 2007 [↑](#footnote-ref-2)