

BAB II

LANDASAN TEORI

II.1. Penelitian Terdahulu

Berikut adalah beberapa jurnal penelitian terdahulu terkait judul penelitian skripsi ini. Berdasarkan penelitian yang dilakukan oleh Suriski Sitinjak (2014) dengan judul penelitian Analisis Untuk Kerja Pelatihan JST Backpropagation Pada Pengenalan Tulisan Tangan Aksara Batak Toba. Aksara Batak Toba merupakan rumpun aksara Batak yang menjadi warisan kebudayaan Indonesia yang patut dilestarikan karena terancam punah akibat keterbatasan sumber data dan informasi. Melalui pemanfaatan perkembangan Teknologi Informasi khususnya di bidang pengenalan tulisan tangan, penelitian ini mengangkat pengenalan tulisan tangan aksara Batak Toba sebagai objek yang menarik dan tepat untuk diteliti. Pengenalan tulisan tangan aksara Batak Toba dilakukan menggunakan metode Wavelet untuk proses dekomposisi dan jaringan saraf tiruan Backpropagation dengan momentum untuk klasifikasinya. Akuisisi data dilakukan secara offline dimana inputan untuk sistem berupa citra huruf tulisan tangan aksara Batak Toba hasil capture menggunakan kamera digital dan diproses untuk diolah oleh sistem. Melalui penelitian ini dapat terlihat kinerja Backpropagation dalam proses pembelajaran pada objek tulisan tangan. Dari pengujian diperoleh penggunaan momentum pada jaringan Backpropagation dapat mempercepat waktu proses pembelajaran dibandingkan pembelajaran tanpa faktor momentum, selain itu unjuk kerja yang dihasilkan juga lebih baik dengan nilai MSE yang mencapai 0.4519 dengan

pengujian pada data yang sama. Hasil penelitian ini juga diharapkan mampu memberikan kontribusi untuk mensosialisasikan bentuk asli aksara Batak Toba dan memperkenalkan serta melestarikan warisan budaya aksara Batak Toba.

Berdasarkan penelitian yang dilakukan Sugeng Winardi (2015) dengan judul penelitian Rancang Bangun Analisis Pengenalan Tulisan Tangan Aksara Hanacaraka, di Indonesia terdapat banyak sekali warisan dan kekayaan budaya. Salah satu warisan budaya bangsa tersebut adalah tulisan/aksara Hanacaraka. Sebagai salah satu warisan kekayaan budaya Indonesia, aksara daerah khususnya aksara Hanacaraka perlu dilestarikan. Warisan budaya tersebut saat ini sudah mulai banyak ditinggalkan oleh masyarakat khususnya masyarakat Jawa dimana aksara Hanacaraka tersebut berada. Pengenalan tulisan tangan aksara Hanacaraka dapat dijadikan untuk objek penelitian dengan cara menerapkan salah satu metode pengenalan tulisan yang ada, misalnya dengan menggunakan metode jaringan saraf tiruan Backpropagation. Sebagai salah satu metode jaringan saraf tiruan Backpropagation terbukti banyak digunakan dan cukup handal untuk pengenalan karakter dan tulisan tangan atau untuk pengenalan citra yang lainnya. Dengan menerapkan metode Backpropagation untuk mengenali pola tulisan tangan aksara Hanaraka, maka dari beberapa contoh tulisan tangan yang berbeda, diharapkan dapat diperoleh hasil akurasi pengenalan yang cukup tinggi. Aplikasi untuk analisis pengenalan tulisan tangan aksara Hanacaraka ini dikembangkan dengan perangkat lunak C#. Hasil dari penelitian ini juga diharapkan mampu untuk ikut melestarikan aksara Hanacaraka sebagai salah satu warisan budaya bangsa Indonesia dengan cara belajar bagaimana menuliskan aksara Hanacaraka tersebut secara benar.

II.2. Landasan Teori

Berikut adalah penjelesan dari beberapa teori yang berhubungan dengan judul pada penelitian ini yaitu :

II.2.1. Analisis

Menurut Anastasia Diana (2012 : 47) Analisis sistem merupakan sekumpulan prosedur untuk membuat spesifikasi sistem informasi yang baru atau sistem informasi di modifikasi. Agar efektif, maka seorang analis sistem harus memiliki pengetahuan dalam bidang komputer, maka di dalam tim pengembangan sistem harus ada orang yang memiliki keahlian dalam bidang bisnis.

Tujuan analisis sistem adalah mengembangkan persyaratan bagi sistem baru. Analisis sistem memerlukan studi terhadap sistem yang ada dan solusi yang diajukan lebih rinci daripada pada tahap survei atau investigasi sistem. Jika survei atau investigasi sistem membantu manajemen menentukan masalah dan membantu memilih apakah akan melanjutkan pengembangan sistem atau tidak, analisis sistem dilakukan untuk memperoleh informasi tambahan yang berguna untuk menjelaskan masalah secara keseluruhan dan memilih serta mengevaluasi solusi masalah, sehingga manajemen dapat memutuskan apakah pengembangan sistem akan dilanjutkan. Apabila pengembangan sistem akan dilanjutkan, maka solusi yang dipakai sudah diketahui dalam analisis sistem ini pilihan dari alternatif solusi sudah dijustifikasi dengan pertimbangan biaya manfaat (*cost benefit*). Selain itu, persyaratan fisik atas disain yang dipilih serta anggaran untuk tahap perancangan sistem juga ditentukan pada tahap ini

II.2.2. Pengertian Aplikasi

Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rahmatillah ; 2011: 3)

II.2.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan adalah model komputasi dari otak manusia. JST merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik baik struktur dan cara kerja menyerupai jaringan syaraf manusia. Hal ini dikarenakan manusia memiliki banyak keunggulan dibandingkan makhluk lain, otak manusia memiliki struktur yang sangat kompleks dan mampu berpikir yang dapat memecahkan persoalan yang dihadapinya dan mampu belajar dari pengalaman masa lalu. Jaringan syaraf biologis pada otak manusia terdiri dari sel-sel syaraf yang disebut *neuron* yang saling berhubungan satu dengan yang lain, pada suatu penghubung yang disebut sinapsis.

Tiga komponen penting pada sel syaraf biologis yang digunakan untuk memahami JST :

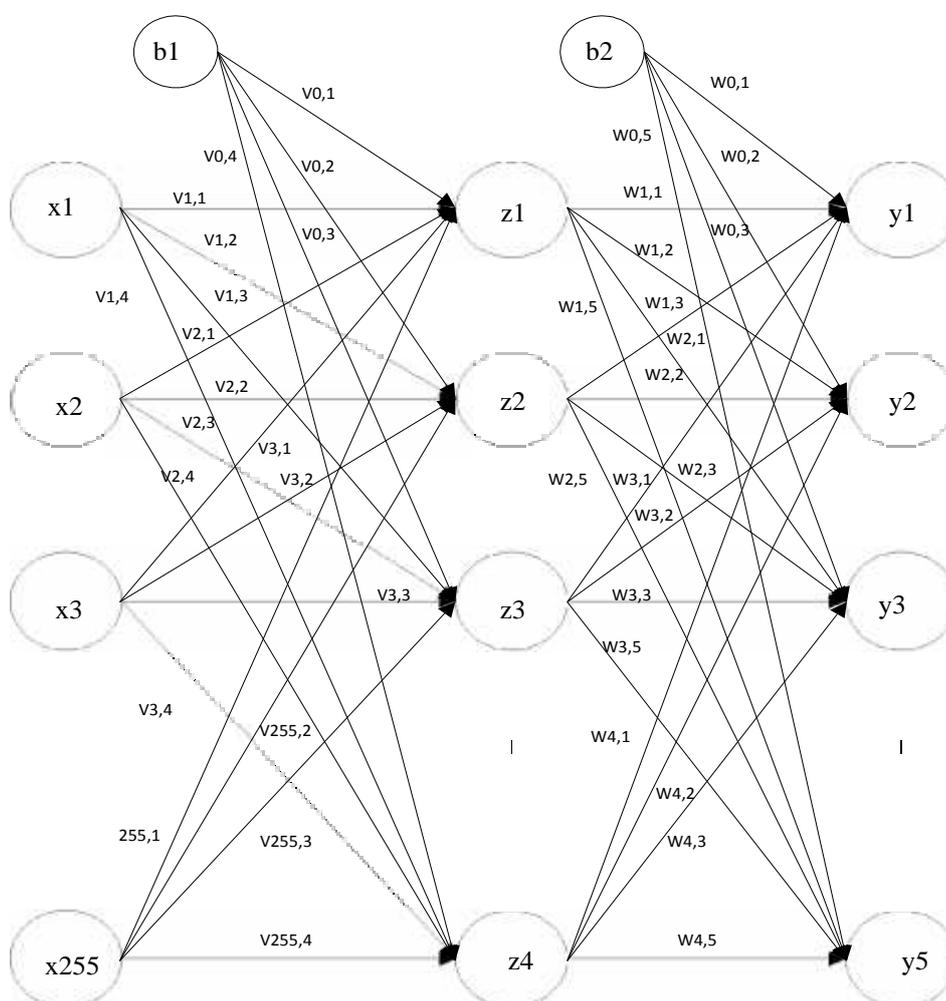
- (1) Dendrit, merupakan elemen pemrosesan yang menerima dan melewatkan sinyal masukan dari neuron lain. Sebuah neuron mampu menerima 5.000 sampai 15.000 sinyal masukan. Sinyal tersebut dimodifikasi dengan bobot (diperkuat/diperlemah) pada sinapsis penerima,
- (2) Soma/badan sel, berfungsi mengakumulasi sinyal masukan terbobot yang dilewatkan melalui dendrit. Jika sinyal – sinyal tersebut lebih besar dari batas ambang tertentu (*threshold*), maka sel akan dipicu sehingga akan mentransmisikan ke neuron lain,
- (3) Akson, berfungsi sebagai saluran keluaran dari suatu neuron yang akan menyalurkan sinyal ke neuron yang lain (Didi Supriyadi ; 2013 : 6)

II.2.3.1. Algoritma *Backpropagation*

Penggunaan dan penerapan metode JST Backpropagation tergolong algoritma pembelajaran/pelatihan yang bersifat supervised dan menggunakan aturan pembelajaran pengoreksian error. Proses pelatihan algoritma/metode *backpropagation* didasarkan pada hubungan yang sederhana, yaitu jika keluaran memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi supaya galatnya dapat diperkecil dan tanggapan JST.

JST Backpropagation memiliki lapisan terdiri dari lapisan input (1 buah), lapisan tersembunyi, dan lapisan output (1 buah). Lapisan input terdiri dari neuron-neuron atau unit-unit input, mulai dari 1 sampai unit input n. Lapisan tersembunyi

(minimal 1). Lapisan tersembunyi terdiri dari unit-unit tersembunyi mulai dari unit tersembunyi 1 sampai unit tersembunyi p. Lapisan output (1 buah). Lapisan output terdiri dari unit-unit output mulai dari unit output 1 sampai unit output m. Nilai n, p, m masing-masing merupakan bilangan integer sembarang yang digunakan metode Backpropagation (Sandy Kosasi ; 2014 : 21).



Gambar II.1 Arsitektur Jaringan Saraf Tiruan

II.2.3.2. Penerapan Metode *Backpropagation*

Prosedur rumus metode *Backpropagation*

Langkah 0 = Inisialisasi bobot (sumainya di atur dengan nilai yang kecil dari 0-1)

Langkah ke 1 = Jika kondisi tidak terapai, lakukan langkah 2 – 12

Langkah ke 2 = untuk setiap pasangan pelatihan, lakukan langkah 3 – 8

Perambanan Maju (Feed Forward)

Langkah ke 3 = Inisiasi bobot dari input layer ke hidden layer

Langkah ke 4 = Hitung hidden layer dengan fungsi sigmoid

$$h = \frac{1}{1+e^{-\sum h \times ih}}$$

Langkah ke 5 = Inisialisasi bobot dari hidden layer ke output layer

Langkah ke 6 = Hitung output layer dengan fungsi sigmoid

$$y = \frac{1}{1+e^{-\sum h \times ho}}$$

Langkah ke 7 = Hitung MSE (Mean Squared Error)

$$\text{MSE} (t - y)^2$$

Perambanan Mundur (Back Propagate)

Langkah ke 8 = Hitung kesalahan dari output ke hidden

$$E_{oh} = y \times (1-y) \times (t - y)$$

Langkah ke 9 = Hitung kesalahan dari hidden ke input

$$\sum E_h = h_o \times E_{oh}$$

$$\text{for } E_{hi} = h \times (1-h) \times \sum E_h$$

Langkah ke 10 = Hitung delta output

$$\text{for } \Delta h_o = E_{oh} \times h$$

Langkah ke 11 = Hitung delta bobot input

$$\text{for } \Delta ih = E_{hi} \times h_i$$

Langkah ke 12 = Perbaiki bobot hidden menuju output

$$ho(\text{baru}) = ho(\text{lama}) + (x \times \Delta ho)$$

Langkah ke 13 = Perbaiki bobot input menuju hidden

$$ih(\text{baru}) = ih(\text{lama}) + (x \times \Delta ih)$$

II.2.3.3. Perhitungan Metode *Backpropagation*.

0100 → 0

Input → Output

1
hidden layer

Tahap pertama (feed forward) dari nilai input 0100

(3) Inisialisasi bobot input ke hidden

Beri nilai acak pada bobot neuron input layer ke hidden layer (sebaliknya di atur dengan nilai yang kecil dari 0 – 1)

(4) hitung hidden layer dengan fungsi sigmoid

$$h = \frac{1}{1+e^{-\sum h_i \times ih_i}}$$

$$\sum h_i \times ih_{i1} = (0 \times 0.2) + (1 \times 0.1) + (0 \times 0.3) + (0 \times 0.3)$$

$$= 0 + 0.1 + 0 + 0$$

$$= 0.1$$

$$h_1 = \frac{1}{1+e^{-0.1}} = 0.525$$

$$\sum h_i \times ih_{i2} = (0 \times 0.5) + (1 \times 0.2) + (0 \times 0.4) + (0 \times 0.5)$$

$$= 0 + 0.2 + 0 + 0$$

$$= 0.2$$

$$h_2 = \frac{1}{1+e} - 0.2 = 0.551$$

- (5) Inisialisasi bobot dari hidden layer ke output layer

$$ho_1 = 0.5 \quad ho_2 = 0.2$$

- (6) Hitung output layer dengan fungsi sigmoid

$$y = \frac{1}{1+e} - \sum h \times h_0$$

$$\sum = (0.525 \times 0.5) + (0.551 \times 0.2) = 0.37$$

$$y = \frac{1}{1+e} - 0.37 = 0.6$$

- (7) Hitung MSE (Mean Squared Error)

$$\text{MSE} (t - y)^2$$

$$= (0 - 0.6)^2$$

$$= -0.6^2$$

$$= 0.36$$

Tahap penambahan mundur (back propogate)

- (8) hitung kesalahan dari output ke hidden

$$E_{oh} = y \times (1-y) \times (t - y)$$

$$= 0.6 \times (1 - 0.6) \times (0 - 0.6)$$

$$= 0.6 \times 0.4 \times -0.6$$

$$= -0.144$$

- (9) hitung kesalahan dari hidden ke ouput

$$\sum E_h = h_0 \times E_{oh}$$

$$= (0.5 \times -0.144) + (0.2 \times -0.144)$$

$$= -0.1008$$

$$E_{hi} = h \times (1 - h) \times \sum E_{hj}$$

$$\begin{aligned} E_{hi_1} &= 0.525 \times (1 - 0.525) \times -0.1008 \\ &= -0.025 \end{aligned}$$

$$\begin{aligned} E_{hi_2} &= 0.551 \times (1 - 0.525) \times -0.1008 \\ &= -0.026 \end{aligned}$$

(10) hitung delta output

$$\Delta h_o = E_{oh} \times h$$

$$\Delta h_{o1} = -0.144 \times 0.525 = -0.0756$$

$$\Delta h_{o2} = -0.144 \times 0.55 = -0.0792$$

(11) hitung delta bobot input

$$\Delta i_h = E_{hi} \times h$$

$$\Delta i_{h_1} = -0.025 \times 0 = 0$$

$$\Delta i_{h_3} = -0.025 \times 1 = -0.025$$

$$\Delta i_{h_5} = -0.025 \times 0 = 0$$

$$\Delta i_{h_7} = -0.025 \times 0 = 0$$

$$\Delta i_{h_2} = -0.026 \times 0 = 0$$

$$\Delta i_{h_4} = -0.026 \times 1 = -0.026$$

$$\Delta i_{h_6} = -0.026 \times 0 = 0$$

$$\Delta i_{h_8} = -0.026 \times 0 = 0$$

(12) perbaiki bobot hidden menunjukkan output

$$h_o (\text{baru}) = h_o (\text{lama}) + (x \times \Delta h_o)$$

$$h_{o_1} (\text{lama}) = 0.5 + (0.4 \times -0.0756)$$

$$= 0.53$$

$$ho_2 \text{ (lama)} = 0.2 + (0.4 \times -0.0792)$$

$$= 0.23$$

(13) perbaiki bobot input menuju hidden layer

$$ih \text{ (baru)} = ih \text{ (lama)} + (x \times \Delta ih)$$

$$ih_1 \text{ (baru)} = 0.2 + (0.4 \times 0) = 0.2$$

$$ih_2 \text{ (baru)} = 0.5 + (0.4 \times 0) = 0.5$$

$$ih_3 \text{ (baru)} = 0.1 + (0.4 \times -0.25) = 0.09$$

$$ih_4 \text{ (baru)} = 0.2 + (0.4 \times -0.26) = 0.096$$

$$ih_5 \text{ (baru)} = 0.3 + (0.4 \times 0) = 0.3$$

$$ih_6 \text{ (baru)} = 0.4 + (0.4 \times 0) = 0.4$$

$$ih_7 \text{ (baru)} = 0.3 + (0.4 \times 0) = 0.3$$

$$ih_8 \text{ (baru)} = 0.5 + (0.4 \times 0) = 0.5$$

II.2.4. Aksara Batak Toba

Aksara Batak Toba merupakan rumpun dari aksara Batak dan bagian dari aksara Nusantara yang menjadi warisan budaya bangsa Indonesia. Pada waktu lampau, aksara Batak Toba tidak digunakan untuk tujuan sehari-hari. Aksara Batak diklasifikasikan sebagai abugida, yaitu paduan antara silabogram (sistem tulisan dengan satuan dasar berupa konsonan yang diikuti oleh sebuah vokal) dan abjad. Aksara Batak Toba terdiri atas dua perangkat huruf, yaitu ina ni surat dan anak ni surat. Agar mudah untuk mengingatnya, urutan yang sering dipakai dalam bentuk kalimat “aha na rata bawa i mangalapa sada gaja” yang artinya “apa yang hijau lelaki itu memotong seekor gajah”. (Suriski Sitinjak ; 2014 : 2)

II.2.5. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (M. Ichwan ; 2011 : 15).

II.2.5.1. Platform Android

Platform Android adalah sebuah *software stack* produksi google untuk perangkat *mobile* yang terdiri atas sistem operasi, *middleware*, dan *key applications*. Aplikasi Android dapat dikembangkan melalui *Android Standart Development Kit* (Android SDK) menggunakan sintaks bahasa pemrograman Java. Aplikasi Android nantinya tidak akan berjalan langsung diatas kernel sistem operasi namun berjalan diatas Dalvik, sebuah *virtual machine* yang khusus dirancang untuk digunakan pada sistem *embedded*. Arsitektur system terdiri atas 5 *layer*, pemisahan *layer* bertujuan untuk memberikan abstraksi sehingga memudahkan pengembangan aplikasi, *Layer-layer* tersebut adalah *layer* aplikasi, *layer framework* aplikasi, *layer libraries*, *layer runtime*, dan *layer kernel* (Wildan Habibi ; 2011 : 2).

II.2.5.2. Seri Android

Berikut adalah versi android dari tahun ke tahun :

1. Android versi 1.1 Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.
2. Android versi 1.5 (Cupcake) Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan system
3. Android versi 1.6 (Donut) Donut (versi 1.6) dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.
4. Android versi 2.0/2.1 (Eclair) Pada 3 Des 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah

pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

5. Android versi 2.2 (Froyo: Frozen Yoghurt) Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market
6. Android versi 2.3 (Gingerbread) Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.
7. Android versi 3.0/3.1 (Honeycomb) Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android

3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8. Android versi 4.0 (ICS: Ice Cream Sandwich) Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.
9. Android versi 4.1 (Jelly Bean) Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus7.
10. Android versi 4.2 (Jelly Bean) Mengusung slogan “smart, simple, and truly yours“, fitur pertama yang diangkat oleh Android 4.4 Kitkat adalah fitur voice search. Meski sudah agak lama memiliki fitur pencarian dengan suara, pada iterasi terbaru ini Google mengimplementasikan aplikasi voice search sebagai

aplikasi yang bisa diakses dengan panggilan saja, tanpa harus menyentuh perangkat sama sekail. Cukup dengan mengucapkan “Ok Google”, maka aplikasi siap untuk melakukan pencarian yang Anda perintahkan. Google juga memberikan beberapa perubahan lain seperti dukungan multitasking yang lebih cepat (Nurul Khotimah ; 2014 : 2).

II.2.6. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem

informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol Use Case

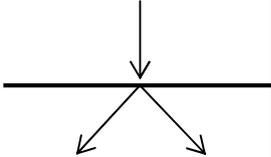
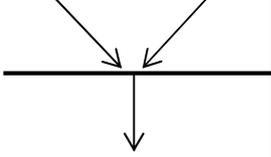
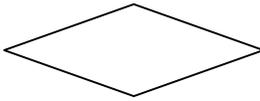
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol Activity Diagram

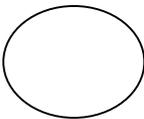
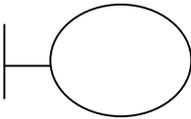
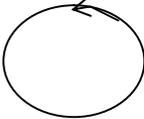
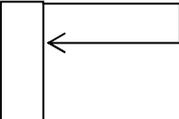
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
 New Swimline	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)