

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Jurnal

Menurut Agus Gede Adi Prayoga dkk (2015:150), menyatakan bahwa: Bahasa Jepang merupakan bahasa yang berbeda dengan bahasa lain pada umumnya karena penulisannya menggunakan huruf Katakana dan Hiragana. Kebutuhan akan bahasa Jepang sangat beragam terutama dalam berkomunikasi contohnya menyambut wisatawan. Pembelajaran bahasa Jepang di Indonesia menemui beberapa kendala diantaranya kurangnya sarana pembelajaran serta suasana belajar yang cenderung membosankan sehingga banyak siswa mengalami kesulitan belajar. Game edukasi pada perangkat mobile merupakan metode pembelajaran baru yang dinilai dapat lebih menarik minat seseorang untuk belajar. Game Edukasi Mengenal Huruf Katakana dan Hiragana dibuat bertujuan membantu mengatasi kesulitan belajar bahasa Jepang terkait penguasaan huruf Katakana dan Hiragana. Materi yang disisipkan dalam game bersumber dari kurikulum Ni Hon Go No Kyoukasho dan Ni Hon Go 1. Game memiliki tiga fitur pembelajaran diantaranya tabel, menulis dan tebak huruf serta fitur permainan sebagai hiburan. Berdasarkan hasil kuesioner 30 siswa yang mengalami kesulitan belajar bahasa Jepang, sebanyak 60% responden menyatakan game mudah dipahami sebagai media pembelajaran bahasa Jepang.

Menurut Sunarti dkk (2016:9), menyatakan bahwa: Pembelajaran bahasa asing merupakan sebuah proses yang dilakukan tahap demi tahap. Berkaitan dengan pembelajaran Bahasa Jepang, penguasaan akan Huruf Hiragana dan Katakana merupakan tahap paling awal yang harus dilewati, baik penguasaan bentuk maupun pengucapan yang benar. Hambatan terjadi oleh karena perbedaan yang cukup mencolok antara kedua huruf tersebut dengan huruf latin yang biasa digunakan oleh masyarakat, sehingga akan berakibat pada minat seseorang untuk mempelajarinya lebih lanjut. Terlebih lagi jika pembelajar tersebut kesulitan mengalokasikan waktunya jika harus mengikuti sebuah sesi kursus bahasa. Selain itu, perlunya arahan pada pembelajar untuk mengetahui urutan penulisan huruf dan pengucapan huruf (suara) yang sesuai dengan masyarakat Jepang asli untuk menghindari kesalahan komunikasi. Untuk itu di dalam penelitian ini dibangun sebuah aplikasi yang dapat mengakomodir kebutuhan tersebut. Aplikasi yang dibangun berbasis Android dan dapat dipasang pada perangkat bergerak yang dimiliki oleh masyarakat. Aplikasi ini di uji pada android 4.1 Jelly Bean dan Android 5.0 Lollipop. Sehingga, pada akhirnya diharapkan minat

masyarakat untuk meneruskan atau bahkan memulai untuk mempelajari sebuah bahasa asing, terutama Bahasa Jepang, menjadi tinggi.

## 2.2 Konsep Dasar Program

### 2.2.1 Java

Menurut Suyanto (2015:2), menyatakan bahwa:

Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari *sun microsystem* yang memiliki kode *green* dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitektur tertentu, mulanya disebut oak, tetapi karena oak sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada, maka *sun* mengubahnya menjadi *java.sun* kemudian meluncurkan *browser* dari *java* yang disebut *hot java* yang mampu menjalankan *applet*. Setelah itu teknologi *java* diadopsi oleh *Netscape* yang memungkinkan program *java* dijalankan di *browser netscape* yang kemudian diikuti *Internet Explore*. Karena keunikan dan kelebihanannya, teknologi *java* mulai menarik banyak vendor seperti *IBM*, *Symantec*, *Inprise*, dll. Sun merilis versi awal *java* secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul *jdk 1.1* kemudian *jdk 1.2* yang mulai disebut sebagai versi *java2* karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah *swing* yang merupakan teknologi *GUI (Graphical User Interface)* yang mampu menghasilkan windows yang portable. Dan pada tahun 1998-1999 lahirlah teknologi *J2EE (Java 2 Enterprise Edition)*.

Java juga memiliki banyak fitur-fitur yang menarik menurut Suyanto (2015:3). Antara lain sebagai berikut:

#### 1. Applet

Program Java yang dapat berjalan di atas browser, yang dapat membuat halaman HTML lebih dinamis dan menarik.

#### 2. Java Networking

Sekumpulan API (Application Programming Interface) yang menyediakan fungsi-fungsi untuk aplikasi – aplikasi jaringan, seperti penyediaan akses untuk TCP, UDP, IP Address dan URL. Tetapi Java Networking tidak menyediakan akses untuk ICMP dikarenakan alasan security dan pada

kondisi umum hanya administrator (root) yang bisa memanfaatkan protokol ICMP.

### 3. Java Database Connectivity (JDBC)

JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server.

### 4. Java Security

Java Security menyediakan sekumpulan API untuk mengatur security dari aplikasi Java baik secara high level atau low level, seperti public/private key management dan certificates.

### 5. Java Swing

Java Swing menyediakan sekumpulan API untuk membangun aplikasi-aplikasi GUI (Graphical User Interface) dan model GUI yang diinginkan bisa bermacam-macam, bisa model Java, model Motif CDE atau model yang independent terhadap platform yang digunakan.

### 6. Java RMI

Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (Remote Procedure Call) jadi object-object Java bisa di call secara remote pada jaringan komputer.

### 7. Java 2D/3D

Java 2D/3D menyediakan sekumpulan API untuk membangun grafik – grafik 2D/3D yang menarik dan juga akses ke printer.

## 8. Java Server Pages

Berkembang dari Java Servlet yang digunakan untuk menggantikan aplikasi– aplikasi CGI, JSP (Java Server Pages) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi Internet.

## 9. JNI (Java Native Interface)

JNI menyediakan sekumpulan API yang digunakan untuk mengakses fungsi– fungsi pada library yang dibuat dengan bahasa pemrograman yang lain seperti C, C++, dan Basic.

## 10. Java Sound

Java Sound menyediakan sekumpulan API untuk manipulasi sound.

## 11. Java IDL + CORBA

Java IDL (Interface Definition Language) menyediakan dukungan Java untuk implementasi CORBA (Common Object Request Broker) yang merupakan model distributed-Object untuk solusi aplikasi besar di dunia networking.

## 12. Java Card

Java Card utamanya digunakan untuk aplikasi – aplikasi pada smart card, yang sederhana wujudnya seperti SIM Card pada handphone.

## 13. JTAPI (Java Telephony API)

Java Telephony API menyediakan sekumpulan API untuk memanfaatkan devices –devices telephone, sehingga akan cocok untuk aplikasi– aplikasi CTI (Computer Telephony Integration) yang dibutuhkan seperti ACD (Automatic Call Distribution), PC-PBX dan lainnya.

### **2.2.2 JDK (*Java Development Kit*)**

Menurut Pangestu dan Widayati (2011:69) mengatakan bahwa:

JDK merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan compiler dan debugger yang diperlukan untuk mengembangkan applet dan aplikasi.

### **2.2.3 SDK (*Software Development Kit*)**

Menurut Safaat H (2015:5) mengatakan bahwa:

SDK (*Software Development Kit*) adalah tool API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java. Android merupakan subset perangkat lunak pada ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu API untuk mulai mengembangkan aplikasi pada platform menggunakan bahasa pemrograman java.

### **2.2.4 Android**

Menurut Juhara (2016:1), menyatakan bahwa:

Android adalah sistem Operasi berbasis linux yang dimodifikasi untuk perangkat bergerak ( *Mobile Devices* ) yang terdiri dari Sistem Operasi, middleware, dan aplikasi-aplikasi utama. Awalnya, Android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dibentuknya Organisasi Open Hanset Alliance pada tahun 2007. Selain google, beberapa nama-nama besar juga ikut serta dalam open handset alliance antara lain: Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel.

Android telah mengalami pembaharuan sejak pertama kali dirilis, rata-rata versi terbaru dari android dirillis setiap 6 bulan. Berikut beberapa jenis versi pembaharuan android, level *Application Programming Interface* (API) dan penjelasannya menurut Juhara (2016:2), Yaitu:

### 1. Cupcake

Android 1.5 (API Level 3) yang dikenal dengan nama kode cupcake ini dirilis pada April 2009 API level 3 membawa perbaikan dari versi sebelumnya dan tentu saja fitur baru. Misalnya dukungan On-screen keyboard, video recording, home screen widget, fitur auto pairing pada bluetooth, kernel linux versi 2.6.7 dan kemampuan memperbaiki filesystem sd card yang rusak.

### 2. Donut

Android 1.6 (API Level 4) yang dirilis pada Oktober 2009 ini merupakan rilis minor dengan fitur utama berupa dukungan teknologi CDMA, gesture, text-to-speech engine, dan fitur pencarian cepat yang memungkinkan anda mencari semua hal seperti kontak, riwayat jelajah internet, boobook, dan lainnya.

### 3. Éclair

Android 2.1 (API Level 7) yang dirilis pada Januari 2010 ini mengusung fitur baru seperti live wallpaper dan beberapa fitur baru seperti android 2.0 misalnya menggunakan banyak akun, kontak cepat, bluetooth 2.1 dan profil bluetooth baru yaitu Object Push Profile (OPP) dan Phone Book Access Profile (PBAP).

### 4. Froyo

Android 2.2 (API Level 8) dirilis pada Mei 2010, fitur baru yang dibawa adalah dukungan OpenGL ES 2.0, instalasi aplikasi penyimpanan eksternal (SD card), dan Android Cloud to Device Messaging yang memungkinkan aplikasi melakukan push messaging dan portable hotspot yang memungkinkan piranti android menjadi hotspot wi-fi untuk berbagi koneksi internet.

## 5. Gingerbread

Android 2.3 (API Level 9 dan 10 ) mengusung beberapa fitur baru, antara lain dukungan banyak kamera, Near Field Communication (NFC), download manager service, dukungan terhadap sensor lain seperti giroskop, dan barometer.

## 6. Honeycomb

Android 3.0 (API Level 11) membawa perubahan besar terutama pada tampilan User Interface (UI), yang berubah 13ahasa13, supaya optimal untuk piranti layar besar seperti tablet. Pada versi ini diperkenalkan fragment, action bar, system clipboard, dan cursor loader.

## 7. Ice Cream Sandwich

Android 4.0 (API Level 14), membawa perubahan besar terutama pada tampilan User Interface (UI) android 3.0 supaya cocok pada layar kecil sehingga memungkinkan aplikasi anda tampak konsisten di tablet maupun ponsel. Fiture baru seperti android baem dan wi-fi direct juga ditambahkan.

## 8. Jellybean

Android 4.1 (API Level 16), adalah rilis minor yang membawa perbaikan fungsionalis dan performa rendering User-Interface (UI). Versi 4.2 (API Level 17) mengusung fiture baru seperti gesture typing dan dukungan multiuser di piranti tablet. Versi 4.3 (API Level 18) membawa perbaikan berupa dukungan Bluetooth low energy dan OpenGL ES 3.0 beberapa fiture baru antara lain

dukungan terhadap 14 bahasa internasional dan penulisan text dua arah (kiri ke kanan atau kanan ke kiri untuk 14 bahasa-bahasa tertentu seperti 14 bahasa arab).

## 9. Kitkat

Android 4.4 (API Level 19), ini mengusung sejumlah perbaikan dan fitur terutama dukungan teknologi Near Field Communication (NFC) melalui host card emulation, pencetakan ke printer nirkabel, webview dengan rendering engine chromium, dan dukungan yang lebih baik bagi piranti yang menggunakan ram rendah.

### 2.2.5 Android Studio

Menurut Juansyah (2015:2), mengemukakan bahwa: Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi android dan bersifat open source atau gratis. Peluncuran android studio ini event Google I/O Convergence untuk tahun 2013. Sejak saat itu, android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Menurut Juansyah (2015:3), android studio memiliki fitur:

1. Projek berbasis pada Gradle Build.
2. Refactory dan pembenahan “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
3. Mendukung proguard and app-signing untuk keamanan.
4. Memiliki GUI aplikasi android lebih mudah.
5. Didukung oleh Google Cloud Platform untuk setiap aplikasi yang dikembangkan dan tidak terikat akan bahasa pemrograman yang digunakan.

### 2.2.6 XML (*Extensible Markup Language*)

Menurut Pangestu dan Widayati (2011:69) mengatakan bahwa: XML adalah singkatan dari eXtensible Markup Language. Bahasa markup adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan, dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. Bahasa markup lain yang populer seperti HTML, menggambarkan kepada browser web tentang bagaimana menampilkan format teks, data, dan grafik ke layar komputer ketika sedang mengunjungi sebuah situs web. XML adalah sebuah bahasa markup yang digunakan untuk mengolah meta data (informasi tentang data) yang menggambarkan struktur dan maksud/tujuan data yang terdapat dalam dokumen XML, namun bukan menggambarkan format tampilan data tersebut. XML adalah sebuah standar sederhana yang digunakan untuk medeskripsikan data teks dengan cara *self-describing* (deskripsi diri). XML juga dapat digunakan untuk mendefinisikan domain tertentu lainnya, seperti musik, matematika, keuangan dan lain-lain yang menggunakan bahasa markup terstruktur.

### 2.3 Metode Algoritma

Menurut Khairul dkk (2017:7), mengemukakan bahwa:

Linear Congruent Method (LCM) merupakan metode pembangkitkan bilangan acak yang banyak digunakan dalam program komputer. LCM diperkenalkan oleh oleh D. H. Lehmer pada tahun 1949.

Pembangkitan bilangan acak dengan menerapkan metode LCM dapat dilihat di bawah ini:

$$X_n = (\alpha (X_{n-1}) + c) \bmod m \dots\dots\dots (1)$$

Dimana :

$\alpha$  : konstanta pengali (  $a < m$  )

$c$  : konstanta pergeseran (  $c < m$  )

$m$  : konstanta modulus (  $> 0$  )

$X_n$ : Bilangan acak ke  $n$  (  $\geq 0, < m$  )

## 2.4 Pengujian Aplikasi

Menurut Ayuliana (2009:1), mengemukakan bahwa:

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Karna itu uji coba *blackbox* memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya.

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori menurut Ayuliana (2009:1), diantaranya:

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan performa.
5. Kesalahan inisialisasi dan terminasi.

Menurut Ayuliana (2009:1), ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana validitas fungsionalnya diuji?
2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu?
4. Bagaimana batasan-batasan kelas data diisolasi?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

## 2.5 Peralatan Pendukung

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

Menurut Nugroho (2010:6), mengemukakan bahwa:

UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut Widodo (2011:10), “Beberapa literature menyebutkan bahwa

UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi.”

Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram kelas (Class Diagram)

Bersifat statis, Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

2. Diagram paket (Package Diagram)

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. Diagram use-case (Use case Diagram)

Bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan sequence (Sequence Diagram)

Bersifat dinamis. Diagram urutan adalah iterasiksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

#### 5. Diagram komunikasi (Communication Diagram)

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

#### 6. Diagram statechart (Statechart Diagram)

Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas.

#### 7. Diagram aktivitas (Activity Diagram)

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

#### 8. Diagram komponen (Component Diagram)

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

#### 9. Diagram deployment (deployment diagram)

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen

yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram, dan sebagainya.

### **2.5.2 UI (*User Interface*)**

Menurut Pressman (2014:312), “*user interface* adalah sebuah mesin komunikasi yang efektif antara manusia dengan komputer”. *User Interface* terdiri dari tampilan gambar serta fungsi – fungsi yang dimana pada C# dikenal dengan objek control.

### **2.5.3 Hiragana, dan Romaji**

*Hiragana* adalah alfabet fonetis dasar Bahasa Jepang. *Hiragana* mewakili setiap bunyi kata pada Bahasa Jepang. Jadi kita bisa menulis, menyusun kata dan kalimat menggunakan huruf Hiragana ini, Dikarenakan Bahasa Jepang tidak menggunakan spasi untuk memisahkan kata demi kata, penggunaan *hiragana* yang terlalu banyak harus dihindari karena dapat membuat susunan huruf tersebut tidak terbaca atau rancu. *Hiragana* juga dapat dipakai untuk menulis nama, entah itu nama orang, binatang, tumbuhan, barang, dan lain – lain. Jumlah huruf *hiragana* ini ada 46 huruf (*Japanese Language Resource* 2009).

*Romaji* mewakili bunyi dari huruf Bahasa Jepang dengan 26 huruf alfabet latin. *Romaji* tidak termasuk ke sistem penulisan Bahasa Jepang, tetapi fungsinya sendiri cenderung untuk menolong orang – orang diluar Jepang

membaca atau mengerti cara pembacaan Bahasa Jepang. *Romaji* sendiri cenderung lebih udah digunakan bagi pemula untuk belajar Bahasa Jepang. Dengan mengerti beberapa kosakata, kita bisa menulisnya dengan *romaji*. *Romaji* digunakan oleh orang Jepang untuk membantu komunikasi dengan orang asing.

#### 2.5.4 Lambang Bunyi pada Huruf *Hiragana*

Selain huruf hiragana dasar, ada pula huruf hiragana yang dimodifikasi dengan menambahkan tanda tertentu atau menggabungkan dengan huruf tertentu sehingga menghasilkan bunyi yang berbeda. Lambang bunyi tersebut antara lain sei-on, daku-on, handaku-on dan yoo-on (Palupi Tria Wardani 2015:15).

##### 1. *Sei-on*

Huruf Hiragana yang berjumlah 46 diatas disebut sei-on. Huruf hiragana yang termasuk sei-on adalah sebagai berikut:

あ	い	う	え	お	/a i u e o/
か	き	く	け	こ	/ka ki ku ke ko/
さ	し	す	せ	そ	/sa shi su se so/
た	ち	つ	て	と	/ta chi tsu te to/
な	に	ぬ	ね	の	/na ni nu ne no/
は	ひ	ふ	へ	ほ	/ha hi fu he ho/
ま	み	む	め	も	/ma mi mu me mo/
や	ー	ゆ	ー	よ	/ya - yu - yo/

Gambar II.1.

Gambar Huruf Hiragana Sei-on

##### 2. *Daku-on*

Bunyi daku-on adalah bunyi yang dihasilkan dari huruf hiragana yang diberi tambahan tanda tenten [“]. Lambang bunyi daku-on berjumlah 20

huruf. Bunyi ini hanya berasal dari set huruf konsonan vokal k, s, t, dan h. Lambang bunyi daku-on adalah sebagai berikut:

が	ぎ	ぐ	げ	ご	/ ga gi gu ge go /
ざ	じ	ず	ぜ	ぞ	/ za ji zu ze zo /
だ	ぢ	づ	で	ど	/ da ji zu de do /
ば	び	ぶ	べ	ぼ	/ ba bi bu be bo /

Gambar II.2.  
Gambar Huruf Hiragana Daku-on

### 3. *Handaku – on*

Huruf Hiragana Daku-oBunyi handaku-on adalah bunyi yang dihasilkan dari huruf hiragana yang diberi tambahan tanda maru [o]. Lambang bunyi handaku-on berjumlah 5 huruf. Bunyi ini hanya berasal dari set huruf konsonan vokal h. Lambang bunyi handaku-on adalah sebagai berikut:

ぱ ひ ぷ ぺ ぽ / pa pi pu pe po/

Gambar II.3.  
Gambar Huruf Hiragana Handaku-on

### 4. *Yoo – on*

Bunyi yoo-on adalah bunyi yang dihasilkan dari huruf hiragana ki, shi, chi, ni, hi, mi, ri, dan huruf hiragana lambang bunyi dakuon gi, ji, pi dan pi dengan menambahkan huruf や(ya), ゆ(yu), よ(yo) yang ditulis lebih kecil daripada ukuran huruf yang seharusnya. Apabila ditulis sama dengan ukuran huruf yang seharusnya maka cara membacanya akan berbeda. Contoh: ひや (hiya) dan ひゃ (hya). Lambang bunyi yoo-on berjumlah 33 huruf. Lambang bunyi yoo-on adalah sebagai berikut:

きゃ きゅ きょ / *kya kyu kyo* /  
 しゃ しゅ しょ / *sha shu sho* /  
 ちゃ ちゅ ちょ / *cha chu cho* /  
 にゃ にゅ にょ / *nya nyu nyo* /  
 ひゃ ひゅ ひょ / *hya hyu hyo* /  
 みゃ みゅ みょ / *mya myu myo* /  
 りゃ りゅ りょ / *rya ryu ryo* /  
 ぎゃ ぎゅ ぎょ / *gya gyu gyo* /  
 じゃ じゅ じょ / *ja ju jo* /  
 びゃ びゅ びょ / *bya byu byo* /  
 ぴゃ ぴゅ ぴょ / *pya pyu pyo* /

Gambar II.4.

Gambar Huruf Hiragana Yoo-on