PEMBANGUNAN PERANGKAT LUNAK UNTUK MENGKONVERSI TEKS AL-QUR'AN KE BRAILLE

Hidayat, M.T.[1], Ahmad Kosasih[2]
[1] Teknik Komputer – Universitas Komputer Indonesia
[2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipati Ukur 112-114 Bandung
E-mail : hidayat@unikom.ac.id[1], ahmad.kosasih@outlook.com[2]

ABSTRAK

Pembangunan perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille ini bertujuan untuk mengamati bagaimana proses pengkonversian karakter Arab beserta tanda bacanya dari teks al-Qur'an menjadi teks Braille serta menganalisis seberapa akurat algoritma yang digunakan untuk pengkonversian teks al-Qur'an ke Braille. Penelitian ini memaparkan bagaimana proses rekayasa perangkat lunak digunakan untuk menyelesaikan permasalahan-permasalahan yang ditemukan dalam pengkonversian teks al-Qur'an ke Braille hingga akhirnya dapat dirancang suatu perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille.

Rancangan perangkat lunak ini menggunakan pemodelan UML dan diimplementasikan dalam bahasa pemrograman Java. Hasil yang diharapkan dari pengkonversian ini adalah kesamaan hasil yang diperoleh dari luaran aplikasi yaitu berupa file teks hasil pengkonversian dengan tampilan yang ada pada mushaf al-Qur'an Braille.

Kata kunci : al-Qur'an, Braille, konversi, perangkat lunak, Java

1. PENDAHULUAN

1.1 Latar Belakang

Dalam bidang teknologi informasi, konversi berkaitan erat dengan perubahan data dari satu bentuk ke bentuk yang lain namun tetap menghasilkan informasi yang sama. Sebagai contoh, pada kasus file audio yang dikonversi dari format wav ke mp3, format file memang berubah akan tetapi suara musik yang dihasilkan tetap sama walaupun ada beberapa nada yang dihilangkan karena dianggap tidak terlalu berpengaruh pada pendengaran manusia.

Saat ini, al-Qur'an Braille hanya terdapat dalam bentuk Braille cetak. Satu buku al-Qur'an Braille cetak hanya memuat satu *juz* al-Qur'an, sehingga 30 *juz* al-Qur'an Braille terdiri dari 30 buku. Kondisi tersebut menyebabkan al-Qur'an Braille tidak praktis untuk dibawa-bawa. Hal ini mendorong untuk dirancangnya sebuah perangkat al-Qur'an Braille elektronik yang dapat menampung 30 *juz* al-Qur'an Braille sekaligus untuk memudahkan tunanetra dalam menggunakan al-Qur'an Braille. Pada tahap awal perancangan, diperlukan perangkat lunak pengkonversi dari teks al-Qur'an ke Braille.

Sebelumnya telah ada perangkat lunak untuk mengkonversi teks dari berbagai bahasa menjadi kode Braille, salah satu diantaranya adalah *Duxbury Braille Translator*. Perangkat lunak ini mampu mengkonversi teks alfabet dari berbagai bahasa kedalam teks Braille. Namun perangkat lunak ini memiliki keterbatasan yaitu tidak dapat mengkonversi teks al-Qur'an secara langsung menjadi teks Braille. Hal ini dikarenakan terdapat aturan khusus tentang cara pengkonversian dan penulisan *Mushaf* al-Qur'an Braille untuk digunakan di Indonesia.

Penelitian ini merupakan bagian dari penelitian yang sudah dilakukan [1]. Sebagai tempat studi kasus, penelitian dilakukan di percetakan *mushaf* al-Qur'an Braille yang berada di Yayasan Wyata Guna. Perluasan yang dilakukan adalah dalam hal data input yang dimasukkan. Hal ini dimaksudkan untuk melihat kemampuan dari perangkat lunak pengkonversi teks al-Qur'an ke Braille yang akan dibangun, sehingga data teks al-Qur'an untuk mesin penampil alphabet Braille dapat diperoleh dengan menggenerasinya dari *database* al-Quran yang sudah ada.

Hasil yang diharapkan dari sebuah pengkonversian adalah kesamaan hasil yang diperoleh dari objek yang dikonversi. Pada kasus pengkonversian teks al-Qur'an ke Braille proses ini jelas membutuhkan keahlian seorang pakar yang sudah berpengalaman di bidang al-Qur'an Braille. Oleh karena itu dibutuhkan suatu penelitian tentang bagaimana merancang suatu perangkat lunak yang mampu memiliki keahlian seperti halnya seorang yang ahli di bidang al-Quran Braille untuk dapat mengkonversi teks al-Qur'an ke Braille.

1.2 Metode Pengumpulan Data

1. Studi Kepustakaan

Kegiatan yang dilakukan pada tahap studi kepustakaan ini adalah menggali informasi tentang perkembangan mushaf al-Qur'an Braille berdasarkan sumber kepustakaan yang diperoleh dari buku-buku, jurnal ilmiah, situs-situs di internet,

paper dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

Salah satu jurnal yang menjadi acuan bagi penelitian ini adalah penggunaan tabel keputusan (decision table) untuk menentukan pilihan tanda baca yang akan ditambahkan pada al-Qur'an Braille [11]. Dimana dalam jurnal tersebut dijelaskan bagaimana cara menambahkan tanda bacaan tajwid dalam al-Qur'an Braille melalui serangkaian tabel keputusan yang berisi kondisi dan aksi yang harus dilakukan jika menemukan kondisi tersebut.

2. Wawancara

Tahap wawancara dilakukan pada bulan Desember 2012 dengan narasumber yaitu kepala bagian percetakan al-Qur'an Braille yang bertempat di yayasan Wyata Guna, Bandung. Setelah kegiatan wawancara dilakukan, maka diperoleh informasi sebagai berikut:

a. Dalam proses belajar membaca al-Qur'an Braille, tunanetra mendengarkan terlebih dahulu ayat al-Qur'an yang akan dibaca beserta tajwidnya kemudian menghafalkan bacaannya. Pada saat membaca al-Qur'an melalui mushaf al-Qur'an Braille, tunanetra melantunkan bacaan ayat yang telah dihafal tadi sambil meraba mushaf al-Qur'an Braille untuk mencocokkan posisinya dengan ayat yang dibaca.
b. Pada mushaf al-Qur'an Braille seringkali terdapat perbedaan cara penulisan tulisan pada kata-kata tertentu dalam al-Qur'an hal ini disebabkan karena adanya dua metode penulisan kata yang digunakan dalam bahasa Arab yaitu kaidah imlaiyah dan kaidah rasm utsmani.
c. Cetakan mushaf al-Qur'an Braille untuk orang awas (yang dapat melihat) baru tersedia 5 juz mulai dari juz 1 hingga juz 5. Jika ingin membandingkan tampilan yang ada di al-Qur'an standar dengan al-Qur'an Braille hanya bisa dilakukan hingga juz ke-5.

3. Observasi

Teknik pengumpulan data yang dilakukan dengan cara melakukan penelitian secara langsung ke objek penelitian sehingga didapat data yang akurat dan lengkap. Pada tahap ini dilakukan pengamatan secara langsung bagaimana al-Qur'an Braille dicetak, format pencetakan, cara membacanya beserta hukum tajwidnya hingga siap digunakan oleh tunanetra. Format untuk cetakan al-Qur'an Braille penulis dapatkan dari mushaf al-Qur'an Braille yang diterbitkan oleh yayasan Wyata Guna. Dalam dokumen tersebut didapatkan format untuk penulisan ayat dalam al-Qur'an Braille adalah 28 huruf perbaris dengan satu spasi di setiap awal ayat.

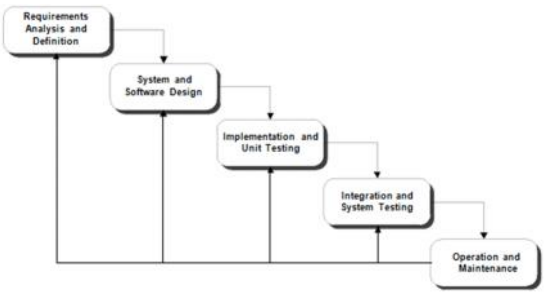
4. Analisis Data

Pada tahap ini dilakukan analisis data masukan yang berupa file teks yang berisi database al-Qur'an berformat xml dan mencocokkan data keluarannya dengan data yang diperoleh dari mushaf al-Qur'an Braille untuk orang awas. Database al-Qur'an yang digunakan seharusnya adalah database al-Qur'an yang datanya sama persis dengan tampilan ayat yang yang ada di mushaf al-Qur'an Standar Indonesia karena yang dibutuhkan oleh pengguna nantinya ketika perangkat lunak ini diimplementasikan adalah tampilan al-Qur'an standar Indonesia. Namun karena tidak adanya database yang dimaksud, maka dicari alternatif lain yaitu menggunakan database al-Qur'an yang telah tersedia di internet yang datanya cukup sama dengan al-Qur'an standar Indonesia.

Setelah melakukan penelitian dengan membandingkan beberapa database al-Qur'an yang telah didapat akhirnya diputuskan untuk menggunakan database al-Qur'an dari situs http://www.tanzill.net [2]. Karena setelah dibandingkan dengan database al-Qur'an yang lain, database tersebut memiliki kesamaan yang cukup akurat dengan al-Qur'an standar Indonesia meskipun masih terdapat beberapa perbedaan dalam segi penempatan tanda waqaf dan penulisan kata-kata tertentu dalam al-Qur'an.

1.3 Metode Pembangunan Perangkat Lunak

Metode yang digunakan dalam pembangunan perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille ini menggunakan paradigma *waterfall* [7]. Berikut ini akan dijelaskan mengenai tahap-tahap pembangunan perangkat lunak ini dengan menggunakan paradigma *waterfall* [gambar 1].



Gambar **Error! No text of specified style in document.**. Metode Waterfall

1. Requirements Analysis and Definition

Pada tahap ini dilakukan analisa terhadap permasalahan dan menetapkan berbagai kebutuhan yang diperlukan dalam pembangunan atau pengembangan suatu perangkat lunak.

Dimulai dari analisis masalah untuk menentukan kebutuhan perangkat lunak, melakukan analisis pada proses pengkonversian secara manual dan mengubahnya ke dalam serangkaian algoritma, pemilihan model yang tepat untuk pemodelan

perangkat lunak hingga pendefinisian *class* dan tipe data.

2. System and Software Design

Tahap menerjemahkan kebutuhan-kebutuhan yang dianalisis ke dalam bentuk yang mudah dimengerti. Sehingga didapat jelas fungsi dan kebutuhan yang diinginkan dari pembangunan atau pengembangan perangkat lunak pengkonversi teks al-Qur'an ke Braille ini. Langkah selanjutnya yang dilakukan adalah merancang tampilan perangkat lunak agar mudah dimengerti oleh pengguna sesuai dengan fungsi dan kebutuhan perangkat lunak pengkonversi teks al-Qur'an ke Braille.

3. Implementation and unit testing

4.

Tahap penerjemahan data atau pemecahan masalah yang telah dirancang kedalam suatu bahasa pemrograman tertentu. Sehingga dihasilkan perangkat lunak pengkonversi teks al-Qur'an ke Braille yang sesuai dengan kebutuhan pengguna. Setiap fungsional yang ada pada perangkat lunak dilakukan uji kelayakan, sehingga perangkat lunak pengkonversi teks al-Qur'an ke Braille ini dapat berjalan dengan baik.

5. Integration and System Testing

Pada tahap ini, dilakukan penyempurnaan terhadap perangkat lunak pengkonversi teks al-Qur'an ke Braille secara keseluruhan agar dapat berjalan sesuai dengan kebutuhan. Pengujian perangkat lunak dilakukan dengan membandingkan data hasil keluaran dari perangkat lunak dengan data yang ada pada *mushaf* al-Qur'an Braille untuk orang awas sehingga diperoleh seberapa besar tingkat akurasinya.

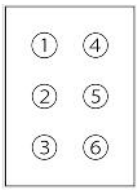
6. Operation and Maintenance

Tahap akhir dimana perangkat lunak pengkonversi teks al-Qur'an ke Braille sudah selesai dapat dioperasikan langsung oleh pengguna. Tahap *maintenance* perlu dilakukan untuk disesuaikan apabila ada perubahan sesuai dengan permintaan pengguna.

2. LANDASAN TEORI

2.1 Pengenalan Huruf Hijaiyah

Huruf *hijaiyah* dalam tulisan Arab Braille memiliki fungsi yang sama dengan tulisan Arab biasa. Perbedaannya terletak pada bentuk huruf dan cara membacanya. Huruf *hijaiyah* dalam Arab Braille berbentuk titik-titik timbul yang berjumlah enam titik pada setiap petaknya dan dibaca dari kiri ke kanan [5]. Untuk lebih jelasnya dapat dilihat pada gambar 2.



Gambar **Error! No text of specified style in document.**. Posisi Titik Pada Huruf Braille

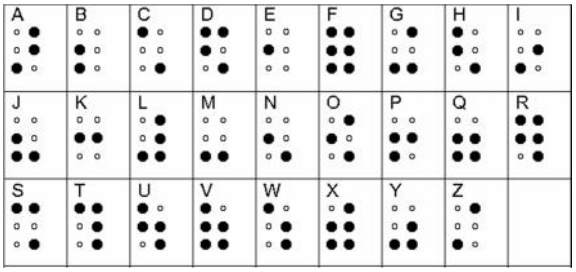
Dari keenam titik tersebut dapat dibentuk huruf Arab Braille seperti pada tabel-tabel dibawah ini.

Tabel 1. Daftar Huruf Pokok

No	Huruf Hijaiyah	Nama Huruf	Kode Braille	Posisi Titik
1.	ا	alif	⠁	(1-- ---)
2.	ب	ba'	⠃	(12- ---)
3.	ت	ta'	⠾	(-23 45-)
4.	ث	tsa'	⠇	(1–456)
5.	ج	jim	⠴	(-2- 45-)
6.	ح	Ía'	⠓	(1-- -56)
7.	خ	kha'	⠯	(1-3 4-6)
8.	د	Dal	⠙	(1–45-)
9.	ذ	dzal	⠹	(-23 4-6)
10.	ر	ra'	⠗	(123 -5-)
11.	ز	za'	⠵	(1-3 -56)
12.	س	sin	⠎	(-23 4--)
13.	ش	syin	⠩	(1–4-6)
14.	ص	Îad	⠮	(123 4-6)
15.	ض	Ìad	⠿	(12- 4-6)
16.	ط	Ïa'	⠞	(-23 456)
17.	ظ	Ða'	⠻	(123 456)
18.	ع	'ain	⠫	(123 -56)
19.	غ	gain	⠛	(12- --6)
20.	ف	fa'	⠋	(12- 4--)
21.	ق	qaf	⠟	(123 45-)
22.	ك	kaf	⠅	(1-3 ---)
23.	ل	lam	⠇	(123 ---)
24.	م	mim	⠍	(1-3 4--)
25.	ن	nun	⠝	(1-3 45-)
26.	و	waw	⠺	(-2- 456)
27.	ه	ha'	⠓	(12- -5-)
28.	لا	lam-alif	⠡	(123 –6)
29.	ء	hamzah mufradah	⠄	(--3 ---)
30.	ي	ya'	⠽	(-2- 4--)
31.	أ	hamzah 'alal alif	⠜	(--3 4---)
32.	ؤ	hamzah 'alal waw	⠬	(12- -56)
33.	ئ	hamzah 'alal ya	⠼	(1-3 456)
34.	آ	hamzah madd	⠣	(--3 45-)
35.	ة	ta' marbutah	⠧	(1-- --6)
36.	ى	alif maqsurah	⠕	(1-3 -5-)

2.2 Sistem Alfabet Braille

Sistem alfabet Braille menggunakan tabel karakter *Unicode* dan ditampilkan menggunakan *font type* Braille AOE.



Gambar 3. Braille AOE Character Mapping

2.3 Pemodelan Perangkat Lunak

Dalam suatu proses pembuatan *software*, analisa dan rancangan merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegosiasikan, dapat dikatakan bahwa tahapan yang dilakukan adalah tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pembangunan *software* yang berbasis *object-oriented* adalah UML.

Alasan mengapa UML digunakan adalah, pertama, *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua, *dynamic modeling*, dapat dipakai untuk pemodelan sistem dinamis dan *real time* [6].

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Tahapan analisis masalah dilakukan terlebih dahulu sebelum tahapan perancangan perangkat lunak. Hal ini dilakukan agar masalah-masalah yang dihadapi dapat diketahui dengan jelas, bagaimana cara kerja algoritma yang digunakan, sampai solusi yang diajukan untuk permasalahan tersebut. Berdasarkan hasil penelitian, masalah-masalah yang sering terjadi pada saat para ahli mushaf al-Qur'an Braille akan melakukan proses pengkonversian adalah sebagai berikut:

1. Penempatan beberapa tanda baca dan waqaf untuk setiap ayat pada mushaf al-Qur'an cetak yang ada di Indonesia seringkali berbeda dengan yang ada pada mushaf al-Qur'an Braille.
2. Belum adanya database al-Qur'an standar Indonesia yang tersedia secara public yang dapat digunakan oleh para ahli al-Qur'an Braille di Indonesia sehingga akan terdapat perbedaan tanda baca yang digunakan pada database al-Qur'an yang tersedia di internet dengan mushaf al-Qur'an yang dicetak di Indonesia.

3.2 Analisis Algoritma

Analisis algoritma digunakan untuk mengetahui alur proses dari algoritma yang digunakan untuk dapat diterapkan ke dalam perangkat lunak. Pada pembangunan perangkat lunak ini, algoritma merupakan alur logika atau serangkaian aturan yang digunakan oleh para ahli yang telah berpengalaman di bidang al-Qur'an Braille untuk menerjemahkan atau mengkonversi setiap ayat al-Qur'an menjadi susunan huruf Braille yang apabila dibaca akan sama bunyi bacaannya sesuai dengan bunyi bacaan yang ada dalam mushaf al-Qur'an standar.

Teks al-Qur'an yang akan dikonversi tidak dapat dikonversi secara langsung tetapi perlu disesuaikan terlebih dahulu. Fungsi penyesuaian teks adalah untuk menyesuaikan kombinasi teks Arab pada ayat al-Qur'an yang akan dikonversi. Langkah-langkah penulisan teks al-Qur'an Braille berdasarkan kaidah yang telah ditetapkan [4] secara garis besar adalah sebagai berikut:

1. Tulis huruf Arab. Misalnya بmaka tulis ⠃.
2. Tulis tanda baca jika ada. Misalnya terdapat kasrah, maka tulis ⠭ setelah huruf ⠃ menjadi ⠃⠭.
3. Jika terdapat syidah maka tulis syidah (⠒) sebelum huruf dan tanda baca. Misalnya kata رَبِّ menjadi ⠗⠘⠭⠒⠃⠭.
4. Tulis tanda mad jika ada misalnya pada kata ٱلرَّحْمٰن dimana terdapat fathah tegak setelah huruf رَ maka ditulis ⠐⠗⠘⠭⠒⠗⠘⠭⠍⠁⠝⠔⠐⠣⠐⠝⠔⠐⠣.
5. Ulangi langkah tersebut hingga semua karakter berhasil disesuaikan.
6. Setelah semua karakter berhasil disesuaikan langkah terakhir adalah mengubah tiap karakter Arab menjadi karakter Braille berdasarkan nilai logika pada tabel 3.1 hingga tabel 3.17.
7. Setiap huruf arab yang ditulis dapat dimodelkan dengan 3 bit karakter Arab yaitu karakter sebagai huruf (characterType), tanda baca ke-1 (diacriticType) dan tanda baca ke-2 (diacriticType). Ketiga susunan bit tersebut disimpan dalam suatu array yang disebut buffer.

CharacterType	*DiacriticType*	*DiacriticType*	...
Offset 0	*Offset 1*	*Offset 2*	...

Ba	*Kasra*	...	*Dhad*	*Fatha + Shadda*	*Maddah*	...
Offset 0	*Offset 1*	...	*Offset 19*	*Offset 20*	*Offset 21*	...

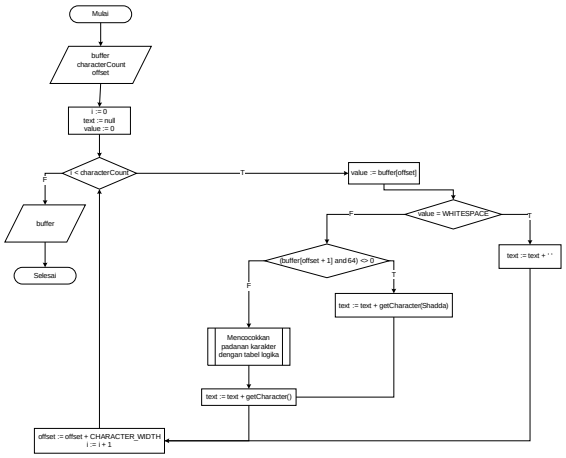
Gambar 4. Pola Susunan Bit pada Karakter Arab dan Contohnya

Posisi untuk setiap huruf, tanda baca dan tanda baca panjang pada setiap karakter arab ditentukan oleh offset. Misalnya untuk tanda kasrah pada huruf ba' seperti contoh pada gambar 3.1 dapat ditemukan pada offset ke-2. Jarak untuk setiap karakter Arab dalam buffer dipisahkan sebanyak 3 bit. Sehingga untuk melompat dari satu karakter Arab ke karakter Arab yang lain harus meloncat sebanyak 3 bit.

Tabel 2. State Tanda_Ayat_Sajdah

State	Tanda_Ayat_Sajdah
Kondisi	buffer[offset] = PlaceOfSajdah
Aksi yang dilakukan	text ← text + '[السجدة]'
Deskripsi	Menambahkan tanda ayat sajdah

Berdasarkan algoritma di atas maka disusunlah flowchart encode character untuk proses penyesuaian karakter Arab pada al-Qur'an Braille sebagai berikut:

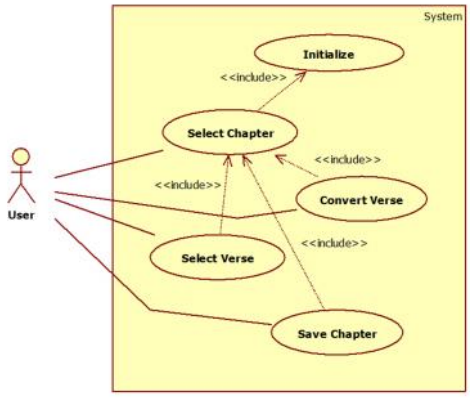


Gambar 5. Flowchart Encode Character

3.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional dianalisis dengan memodelkan perangkat lunak. Pemodelan yang digunakan untuk memodelkan perangkat lunak ini adalah pemodelan berorientasi objek. Perangkat lunak ini dimodelkan menggunakan UML (*Unified Modeling Language*). *Tools* yang digunakan adalah *Use Case Diagram*, *Sequence Diagram*, dan *Activity Diagram*.

Use case diagram perangkat lunak untuk mengkonversi teks al-Qur'an memiliki 1 aktor dan 4 *use case* dimana keempat *use case* ini mewakili setiap fungsionalitas utama pada perangkat lunak. *Use case diagram* perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille ini dapat dilihat pada gambar 3.2.



Gambar 6. Use Case Diagram

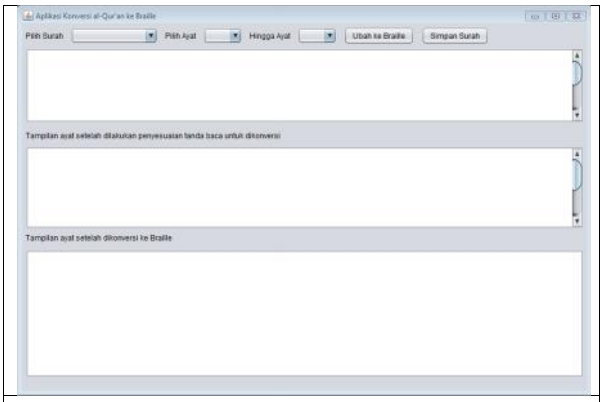
a. Use case initialize merupakan fungsionalitas untuk inisialiasi aplikasi.
b. Use case select chapter merupakan fungsionalitas untuk memilih surah.
c. Use case select verse merupakan fungsionalitas untuk memilih ayat.
d. Use case convert verse merupakan fungsionalitas untuk mengkonversi ayat ke teks Braille.
e. Use case save chapter merupakan fungsionalitas untuk menyimpan hasil konversi teks Braille ke file teks.

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Perangkat Lunak

Tahap implementasi merupakan tahap pemasangan perangkat lunak supaya siap untuk dioperasikan dan dapat dipandang sebagai usaha untuk mewujudkan perangkat lunak yang telah dirancang.

Berikut adalah tampilan antarmuka pada perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille.



Keterangan:
1. Klik daftar *surah* untuk memilih *surah*
2. Klik daftar ayat untuk memilih ayat
3. Klik button ubah ke braille untuk mengkonversi ayat
4. Klik button simpan ke file untuk menyimpan ayat hasil konversi

Gambar 7. Form Antarmuka Aplikasi

4.2 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille ini dapat dilihat pada tabel 4.4

Tabel 3. Rencana Pengujian

No	Proses	Jenis Pengujian
1.	Select Chapter	White box and Black box testing : sample testing
2.	Select Verse	White box and Black box testing : sample testing
3.	Convert Verse	White box and black box testing : sample testing

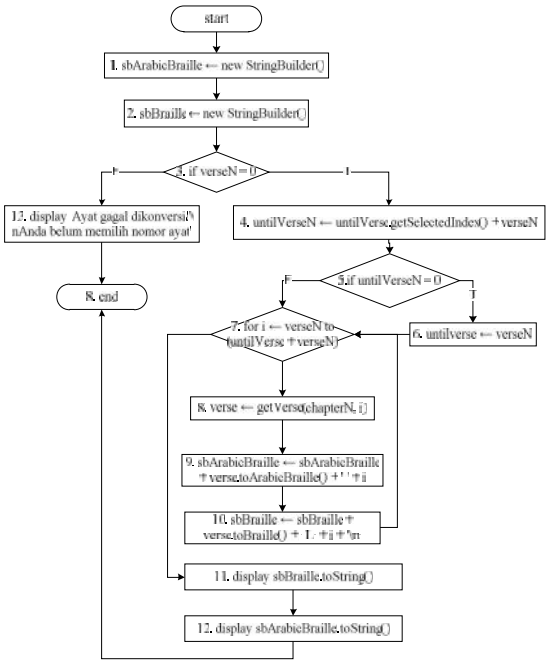
4.	Save Chapter	White box and Black box testing : sample testing

4.3 Pengujian White Box

Pengujian white box bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak apakah berjalan dengan baik atau tidak. Pengujian white box akan digunakan pada algoritma yang digunakan pada aplikasi pengkonversi teks al-Qur'an ke Braille, untuk mengukur kinerja logika berdasarkan pseudocode yang telah dibuat pada tahap analisis.

Pengujian white box dilakukan dengan beberapa tahap, yaitu :

1. Langkah pertama ubah pseudocode menjadi flowchart.
2. Ubah source code menjadi flowchart.
3. Ubah flowchart menjadi flowgraph ke dalam bentuk yang lebih sederhana.
4. Tahap pengujian, dimana tahap pengujian ini dilakukan dengan 5 cara yaitu, menghitung region, menghitung cyclomatic complexity, menghitung independent path, menggunakan graph matriks, menghitung predicate node.



Gambar 8. Pengubahan Source Code Menjadi Flowchart Convert Verse

4.4 Pengujian Pengguna

Pengujian dilakukan dengan melakukan demo program secara langsung, mengoreksi hasil konversinya serta melakukan wawancara dengan beberapa pakar al-Qur'an Braille. Berdasarkan hasil wawancara dengan beberapa pakar al-Qur'an Braille didapatkan kesimpulan sebagai berikut:

1. Perangkat lunak telah mencapai hasil yang diharapkan dari segi fungsionalitas dan hasil keluarannya dapat diaplikasikan pada perangkat keras penampil al-Qur'an Braille.
2. Kesalahan pengkonversian yang disebabkan karena adanya perbedaan cara penulisan yang terdapat pada beberapa kata di al-Qur'an serta penempatan tanda waqaf adalah hal yang wajar, mengingat al-Qur'an Braille diperuntukkan agar mudah dibaca oleh tunanetra sehingga ada penyingkatan pada beberapa kata agar mudah ditulis dalam huruf Braille.

4.5 Rekapitulasi Hasil Pengujian

Tabel **Error! No text of specified style in document.**. Rekapitulasi Hasil Pengujian

No	Pengujian	Jumlah Sampel yang Diuji	Jumlah Sampel yang berhasil Dikonversi	Persentasi Keberhasilan
1.	Tanda Baris Pendek	15	15	100%
2.	Tanda *Tanwin*	10	10	100%
3.	Tanda Baris *Mad*	5	5	100%
4.	Huruf *Mad*	6	6	100%
5.	Tanda *Mad* Bendera	7	5	71%
6.	Huruf *Hamzah*	23	23	100%
7.	Huruf *Lam Alif*	5	5	100%
8.	Huruf *Ta' Marbutah*	5	5	100%
9.	Huruf *Alif Maqsurah*	5	5	100%
10.	Huruf *Alif* dan *Lam*	10	10	100%
11.	Tanda *Tanwin Wasal*	5	5	100%
12.	Huruf *Zaidah*	4	4	100%
13.	Tanda *Waqaf*	18	18	100%
14.	Tanda Ayat *Sajdah*	15	15	100%
15.	Tanda *Saktah*	4	4	100%
	Total	137	135	

Hasil persentase keberhasilan untuk pengkonversian pada 15 kombinasi huruf dan tanda baca yang ada dalam al-Qur'an adalah sebagai berikut:

%Keberhasilan =

$$\frac{\text{Pembacaan huruf/tanda baca yang berhasil dikonversi} * 100\%}{\text{Seluruh pembacaan huruf/tanda baca yang diuji}}$$

$$\frac{135}{137} * 100\% = 98.5\%$$

Dari hasil pengujian dapat dilihat bahwa algoritma dapat mengenali kelima belas kombinasi huruf dan tanda baca yang ada dalam al-Qur'an dan berhasil mengkonversinya dengan baik menjadi luaran yang diinginkan dengan tingkat kerberhasilan **98.5%.**

5. KESIMPULAN

Berdasarkan hasil analisa dan pembahasan yang telah diuraikan, maka pada penelitian skripsi ini dapat ditarik beberapa kesimpulan antara lain:

1. Pembangunan perangkat lunak untuk mengkonversi teks al-Qur'an ke Braille dengan tujuan untuk mengamati proses pengkonversian karakter Arab beserta tanda bacanya dari teks al-Qur'an menjadi teks Braille telah berhasil dilakukan dengan melakukan pengujian pada 15 kombinasi huruf dan tanda baca yang ada dalam al-Qur'an.
2. Berdasarkan data hasil pengujian, algoritma konversi berhasil mengenali 15 kombinasi huruf dan tanda baca yang sering digunakan serta berhasil mengkonversinya sesuai target luaran yang ada di al-Qur'an Braille. Namun masih terdapat error yang disebabkan oleh perbedaan cara penulisan huruf dan tanda baca yang digunakan.
3. Dalam proses pengkonversian terdapat beberapa kondisi yang belum diterapkan dengan sempurna sesuai dengan tampilan yang ada di al-Qur'an Braille.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih pada Direktorat Penelitian dan Pengabdian pada Masyarakat Pendidikan Tinggi dan Universitas Komputer Indonesia yang telah membiayai dan memfasilitasi kegiatan penelitian ini. Dan juga ucapan terima kasih penulis sampaikan pada Percetakan al-Quran Braille Wyata Guna Bandung yang telah memberikan banyak informasi tentang al-Quran Braille khususnya al-Quran Braille standar yang digunakan bagi penyandang tunanetra di Indonesia.

DAFTAR PUSTAKA

[1] Haryadi, Y. (2012). *Rancang Bangun Perangkat Keras Penampil al-Qur'an Braille*. Bandung: Skripsi Universitas Komputer Indonesia.

[2] Anonim. (2010, Desember). *Download Quran Text*. Dipetik Mei 13, 2013, dari Tanzil Quran: http://tanzil.net/download/

[3] Anonim. (2011, Mei). *Java API - Download*. Dipetik Mei 28, 2013, dari The Quranic Arabic Corpus: http://corpus.quran.com/java/download.jsp

[4] Anonim. (2011). *Pedoman Membaca dan Menulis al-Qur'an Braille*. Jakarta: Lajnah Pentashihan Mushaf al-Qur'an.

[5] Anonim. (2005). *Braille AOE Font*. Dipetik Mei 2, 2013, dari Dafont: http://www.dafont.com/braille-aoe.font

[6] Nugroho, A. (2009). *Rekayasa Perangkat Lunak Menggunakan UML dan Java. Yogyakarta*: Andi.

[7] Sommerville, I. (2004). *Software Engineering: Seventh Edition*. Pearson Education.

[8] Guritno, S., Sudaryono, & Raharja, U. (2011). *Theory and Application of IT Research*. Yogyakarta: Andi.

[9] Anonim. (2012). *Al-Qur'anul Karim dalam Huruf Arab Braille Berpedoman Kepada Mushaf Standar*. Jakarta: Lajnah Pentashihan Mushaf al-Qur'an.

[10] O'Docherty, M. (2005). *Object-Oriented Analysis & Design, Understanding System Development with UML 2.0*. British: John Willey & Sons.

[11] Abualkishik, A. M., & Omar, K. (2010). *Quran Vibrations in Braille Codes Using the Finite State Machine Technique.* University Kebangsaan Malaysia,4-6.