



**IMPLEMENTASI ALGORITMA *KNUTH MORRIS*
PRATT PADA PERANCANGAN GAME
HANACARAKA**

Skripsi

Disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Pendidikan
Program Studi Teknik Informatika dan Komputer

Oleh

Abi Mahan Zaky
NIM. 5302410208

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2015**

PERNYATAAN

Saya menyatakan bahwa yang tertulis di dalam skripsi ini benar-benar hasil karya sendiri, bukan jiplakan dari karya tulis orang lain, baik sebagian atau seluruhnya. Pendapat atau temuan orang lain yang terdapat dalam skripsi dirujuk berdasarkan kode etik ilmiah.

Semarang, 28 April 2015



Abi Mahan Zaky

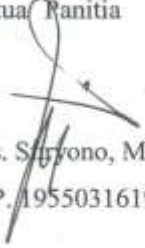
NIM. 5302410208

PENGESAHAN

Skripsi dengan judul “Implementasi Algoritma *Knuth Morris Pratt* pada Perancangan *Game* Hanacaraka” telah dipertahankan di hadapan sidang Panitia Ujian Skripsi Fakultas Teknik Universitas Negeri Semarang pada tanggal 23 Februari tahun 2015

Panitia:

Ketua Panitia


Drs. Suryono, M.T.

NIP. 195503161985031001

Sekretaris


Feddy Setio P., S.Pd., M.T.

NIP. 197808222003121002

Penguji I


Feddy Setio P., S.Pd., M.T.

NIP. 197808222003121002

Penguji II


Aryo Baskoro U., S.T., M.T.

NIP. 198409092012121002

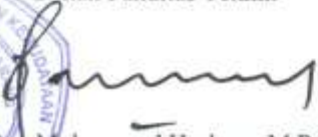
Penguji III/Pembimbing


Riana Defi M. P., S.T., M.T.

NIP. 197609182005012001

Mengetahui
Dekan Fakultas Teknik




Drs. Muhammad Harlanu, M.Pd
NIP. 196602151991021001

MOTTO DAN PERSEMBAHAN

MOTTO:

1. Lebih baik terlambat daripada tidak sama sekali.
2. Bersabar dalam menunggu anugrah, berusaha tanpa kenal lelah
3. Bermanfaat untuk seluruh makhluk dan alam semesta

PERSEMBAHAN:

Skripsi ini peneliti persembahkan untuk:

1. Ibu dan almarhum Bapak tercinta, terima kasih banyak atas semua yang telah diberikan.
2. Adik-adik tersayang yang senantiasa saya sayangi dan banggakan.
3. Teman - teman seperjuangan Pendidikan Teknik Informatika dan Komputer angkatan 2010.
4. Almamaterku Universitas Negeri Semarang.

KATA PENGANTAR

Puji syukur dihaturkan kehadirat Allah swt atas berkah dan rahmatNya sehingga skripsi yang berjudul Implementasi Algoritma *Knuth Morris Pratt* pada Perancangan *Game* Hanacaraka ini dapat diselesaikan. Skripsi ini disajikan sebagai salah satu syarat untuk memperoleh gelar Sarjana Pendidikan Program Studi Pendidikan Teknik Informatika dan Komputer.

Rasa hormat dan terima kasih diucapkan kepada berbagai pihak yang telah membantu dalam penyusunan skripsi ini, terutama kepada:

1. Ibu dan almarhum Bapak tercinta, terima kasih banyak atas semua yang telah diberikan.
2. Yth. Riana Defi Mahadji P., S.T., M.T. selaku Dosen Pembimbing yang telah memberikan bimbingan dan saran selama penulisan skripsi ini
3. Seluruh dosen, karyawan, dan keluarga besar Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang
4. Semua pihak yang telah membantu yang tidak dapat disebutkan satu per satu.

Semoga skripsi ini dapat memberikan manfaat bagi kemajuan ilmu pengetahuan dan teknologi. Aamiin.

Semarang, April 2015



Penulis

ABSTRAK

Zaky, Abi Mahan 2015. *Implementasi Algoritma Knuth Morris Pratt pada Perancangan Game Hanacaraka*. Skripsi, Jurusan Teknik Elektro Fakultas Teknik, Universitas Negeri Semarang. Riana Defi M. P., S.T., M.T.

Kata Kunci : Hanacaraka, *Game* , Algoritma *Knuth Morris Pratt*, Android.

Aksara Jawa merupakan salah satu peninggalan budaya yang tak ternilai harganya. Bentuk aksara dan seni pembuatannyapun menjadi peninggalan yang patut untuk dilestarikan. Aksara ini menjadi bukti nyata adanya zaman terdahulu sebelum adanya bangsa Indonesia. Salah satu upaya melestarikannya adalah dengan melakukan digitalisasi Aksara Jawa, sehingga bangsa Indonesia tidak akan kehilangan nilai budayanya. Perkembangan teknologi terutama teknologi *game* merupakan salah satu teknologi yang dapat dimanfaatkan untuk melestarikan Aksara Jawa ini. Selain itu dapat juga dimanfaatkan untuk menjadi media pembelajaran Aksara Jawa.

Penelitian ini menggunakan metode *Interactive Multimedia System Design and Development* (IMSDD) yang digunakan untuk merancang perangkat lunak *game* Hanacaraka. Metode ini merupakan pengembangan dari metode *waterfall* yang diadaptasi sesuai dengan kebutuhan pengembangan aplikasi multimedia, sehingga metode ini merupakan metode yang paling cocok dalam pengembangan *game* Hacaraka.

Algoritma *Knuth Morris Pratt* adalah salah satu algoritma yang digunakan pada pencarian string. Dalam penelitian ini algoritma *Knuth Morris Pratt* diimplementasikan pada proses pencarian *string* yang terdapat pada *gameplay* utama *game* Hanacaraka.

Hasil penelitian uji fungsionalitas menunjukkan bahwa *game* Hanacaraka, dari semua kriteria penilaian, termasuk dalam kategori sangat layak dengan prosentase 97,14%. Observasi dari implementasi algoritma *Knuth Morris Pratt* pada *game* Hanacaraka menunjukkan bahwa algoritma *Knuth Morris Pratt* berhasil di implementasikan pada *game* Hanacaraka dengan tingkat keberhasilan 100%. Secara keseluruhan, implementasi algoritma *Knuth Morris Pratt* pada perancangan *game* Hanacaraka berhasil dilakukan dan mendapat hasil yang memuaskan dari segi fungsionalitas.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERNYATAAN.....	ii
PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
KATA PENGANTAR	v
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
DAFTAR LISTING PROGRAM	xi
DAFTAR LAMPIRAN.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	3
1.5 Tujuan Penelitian	3
1.6 Manfaat Penelitian	4
1.7 Penegasan Istilah	4
BAB 2 LANDASAN TEORI	6
2.1 Game	6
2.1.1 Elemen <i>Game</i>	6
2.1.2 Genre <i>Game</i>	9
2.1.3 <i>Word Puzzle Game</i>	11
2.2 Android	11
2.2.1 Sistem Operasi Android	11
2.2.2 Arsitektur Android	13

2.3	Algoritma <i>String Matching</i>	15
2.4	Algoritma <i>Knuth Morris Pratt</i>	16
2.5	<i>Interactive Multimedia System Design and Development</i>	17
2.6	<i>Unified Modeling Language (UML)</i>	20
2.7	Kerangka Berfikir	21
BAB 3	METODE PENELITIAN	23
3.1	Waktu dan Tempat Pelaksanaan	23
3.2	Jenis Penelitian dan Desain Penelitian	23
3.2.1	<i>System Requirement</i>	23
3.2.2	<i>Design Consideration</i>	26
3.2.2.1	Analisis Permainan	26
3.2.2.2	Analisis Algoritma <i>Knuth Morris Pratt</i>	34
3.2.2.3	Perancangan Sistem	44
3.2.2.4	Uji Coba Sistem	54
3.3	Teknik Pengumpulan Data	55
3.4	Teknik Analisis Data.....	56
BAB 4	HASIL PERANCANGAN DAN PEMBAHASAN	59
4.1	Implementasi	59
4.1.1	Implementasi dari Algoritma <i>Knuth Morris Pratt</i>	59
4.1.2	Implementasi dari Perancangan Antarmuka	68
4.2	<i>Evaluation</i>	73
4.2.1	<i>Black Box Testing</i>	74
4.2.2	Metode <i>Checklist</i> (Angket)	79
BAB 5	PENUTUP	82
5.1	Kesimpulan	82
5.2	Saran-saran	82
	DAFTAR PUSTAKA	84
	LAMPIRAN	86

DAFTAR TABEL

	Halaman
Tabel 3.1 Spesifikasi Toshiba Satellite C640	24
Tabel 3.2 Spesifikasi SONY Xperia S (LT26i)	25
Tabel 3.3 Skenario Pengujian <i>Blackbox</i>	55
Tabel 3.4 Kategori Predikat Fungsionalitas Perangkat Lunak	58
Tabel 4.1 Pengujian <i>Expected Input</i> pada Perangkat Lunak.....	74
Tabel 4.2 Pengujian <i>Illegal Entry</i> pada Perangkat Lunak	75
Tabel 4.3 Pengujian <i>Boundary Value Analysis</i> pada Perangkat Lunak .	76
Tabel 4.4 Hasil Pengujian <i>Expected Input</i> pada Perangkat Lunak	77
Tabel 4.5 Hasil Pengujian <i>Illegal Entry</i> pada Perangkat Lunak	78
Tabel 4.6 Hasil Pengujian <i>Boundary Value Analysis</i> pada Perangkat Lunak	79
Tabel 4.7 Pengujian Metode Angket pada Perangkat Lunak	80
Tabel 4.8 Hasil Pengujian Metode Angket pada Perangkat Lunak	80
Tabel 4.9 Data Pengujian Metode Angket pada Perangkat Lunak	81

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Elemen Dasar <i>Game</i>	7
Gambar 2.2 Arsitektur Android	13
Gambar 2.3 Siklus IMSDD	19
Gambar 2.4 Kerangka Berfikir	22
Gambar 3.1 <i>Use Case Diagram</i> Hanacaraka	31
Gambar 3.2 <i>Activity Diagram</i>	33
Gambar 3.3 <i>Flowchart</i> Algoritma <i>Knuth Morris Pratt</i>	43
Gambar 3.4 <i>Flowchart</i> Aplikasi Permainan Hanacaraka	45
Gambar 3.5 Rancangan <i>Scene</i> Menu	46
Gambar 3.6 Rancangan <i>Scene</i> Pilih Level	48
Gambar 3.7 Rancangan <i>Scene</i> Pilih Soal	49
Gambar 3.8 Rancangan <i>Scene</i> Arena Permainan	51
Gambar 3.9 Rancangan <i>Scene</i> Cara Main	53
Gambar 4.1 Jawaban Benar	64
Gambar 4.2 Jawaban Salah	65
Gambar 4.3 Animasi <i>Game Over</i>	66
Gambar 4.4 Jawaban Hampir Benar	67
Gambar 4.5 Tampilan <i>Scene</i> Menu	69
Gambar 4.6 Tampilan <i>Scene</i> Pilih Level	70
Gambar 4.7 Tampilan <i>Scene</i> Pilih Soal	71
Gambar 4.8 Tampilan <i>Scene</i> Arena Permainan	72
Gambar 4.9 Tampilan <i>Scene</i> Cara Main	73

DAFTAR LISTING PROGRAM

	Halaman
Listing 3.1 <i>Pseudocode</i> Algoritma KMP fase pra-pencarian	35
Listing 3.2 <i>Pseudocode</i> Algoritma KMP fase pencarian	35
Listing 4.3 <i>Code</i> program algoritma KMP	61

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Kelengkapan	86
Lampiran 2. <i>Form</i> Angket (<i>Checklist</i>)	90
Lampiran 3. Contoh Tabel Pengkoreksian Kata	91

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi informasi saat ini, industri *game* merupakan salah satu industri yang berkembang dengan pesat dan menarik perhatian masyarakat luas. Saat ini, berbagai jenis *game* baik yang *online* maupun *offline* dapat dengan mudah diakses oleh masyarakat melalui warnet, perangkat komputer pribadi, ataupun telepon seluler. Memainkan *game* pun memiliki kelebihan seperti terapi penyakit kronis pada anak, meningkatkan psikomotorik anak, mengurangi stress dan depresi, penghilang rasa sakit, meningkatkan ketelitian, meningkatkan ketrampilan dalam mengambil keputusan, dan membuat tetap bisa gembira di usia tua (Gallagher, 2013).

Pada awalnya konsep *game* dibuat hanya untuk sekedar hiburan bagi masyarakat. Namun seiring berkembangnya waktu, *game* semakin bervariasi dan menarik karena fitur-fiturnya yang selalu berkembang dan memberikan hal-hal baru yang sebelumnya belum pernah ada. Salah satu dari jenis *game* yang berkembang seiring berjalannya waktu adalah jenis *word game* (permainan kata). Dari sekian banyak jenis permainan kata ada satu jenis permainan kata yang cukup populer, yaitu *game* tebak kata.

Aksara Jawa merupakan salah satu peninggalan budaya yang tak ternilai harganya. Bentuk aksara dan seni pembuatannyapun menjadi peninggalan yang patut untuk dilestarikan. Aksara ini menjadi bukti nyata adanya zaman terdahulu

sebelum adanya bangsa Indonesia. Salah satu upaya melestarikannya adalah dengan melakukan digitalisasi Aksara Jawa, sehingga bangsa Indonesia tidak akan kehilangan nilai budayanya.

Perkembangan teknologi terutama teknologi *game* merupakan salah satu teknologi yang dapat dimanfaatkan untuk melestarikan Aksara Jawa ini. Selain itu dapat juga dimanfaatkan untuk menjadi media pembelajaran Aksara Jawa.

Berdasarkan pada pemikiran inilah penulis merasa perlu dibuat *game* yang dapat menjadi media untuk mengenalkan Aksara Jawa yang mengadopsi *genre word game*. Dengan adanya *game* ini diharapkan dapat memudahkan orang-orang yang ingin mempelajari Aksara Jawa. Pada jenis permainan ini terdapat beberapa permasalahan, salah satu diantaranya adalah pencocokan jawaban dari pemain dengan jawaban benar dari *game* tersebut. Salah satu algoritma yang digunakan pada pencarian *string* adalah algoritma *Knuth Morris Pratt*. Oleh karena itu, penulis mengangkat judul “Implementasi Algoritma *Knuth Morris Pratt* pada Perancangan Game Hanacaraka”.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dikemukakan diatas, maka dapat diidentifikasi masalah yang akan dibahas pada penelitian ini yaitu bagaimana mengimplementasikan algoritma *Knuth Morris Pratt* pada perancangan *game* Hanacaraka.

1.3 Batasan Masalah

Pada perancangan implementasi algoritma *Knuth Morris Pratt* pada *game* Hanacaraka terdapat pembatasan masalah pada penelitian yang dilakukan, adalah implementasi algoritma *Knuth Morris Pratt* pada *game* Hanacaraka pada bagian pencocokan *string* jawaban yang dimasukkan pemain dengan jawaban yang ada pada *database game*.

1.4 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka muncul permasalahan bagaimana merancang *game* Hanacaraka sebagai media pengenalan aksara jawa bagi orang umum yang mengimplementasikan algoritma *Knuth Morris Pratt* untuk meningkatkan kualitas sistem pencocokan string pada *game*?

1.5 Tujuan Penelitian

Berdasarkan permasalahan yang telah diungkapkan di atas, maka dapat dirumuskan tujuan dari penelitian yang akan dilakukan. Tujuan dari penelitian ini adalah untuk merancang *game* Hanacaraka sebagai media pengenalan aksara jawa bagi orang umum dengan mengimplementasikan algoritma *Knuth Morris Pratt* dalam sistem pencocokan string pada *game*.

1.6 Manfaat Penelitian

Manfaat yang diharapkan dari hasil penelitian ini dari sisi keilmuan adalah memberikan model *game* yang dapat menjadi media pengenalan aksara Jawa bagi orang umum dengan mengimplementasikan algoritma *Knuth Morris Pratt* pada pencocokan *string*-nya.

1.7 Penegasan Istilah

Berikut ini terdapat beberapa istilah yang berkaitan dengan penelitian ini. Hal ini dimaksudkan agar tidak terjadi salah pengertian. Beberapa istilah yang perlu dijelaskan adalah:

1. *Game*

Dalam kamus bahasa Indonesia *Game* adalah permainan. Permainan merupakan bagian dari bermain dan bermain juga bagian dari permainan keduanya saling berhubungan. Permainan adalah kegiatan yang kompleks yang didalamnya terdapat peraturan, *play* dan budaya. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Penggunaan istilah *game* pada penelitian ini merujuk pada permainan digital atau *game* berbentuk digital yang dapat dimainkan dengan alat elektronik seperti komputer maupun *smartphone*.

2. Algoritma *Knuth Morris Pratt*

Algoritma *Knuth Morris Pratt* adalah salah satu algoritma pencarian *string*, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977.

3. Hanacaraka

Dalam Kamus Besar Bahasa Indonesia (KBBI) Hanacaraka adalah abjad Jawa dan Sunda yang berjumlah dua puluh lambang (disebut *ha, na, ca, ra, ka, da, ta, sa, wa, la, pa, dha, ja, ya, nya, ma, ga, ba, tha, nga*). Pada penelitian ini penggunaan kata Hanacaraka merujuk kepada abjad Jawa yang berjumlah dua puluh lambang tersebut.

BAB II

LANDASAN TEORI

2.1 Game

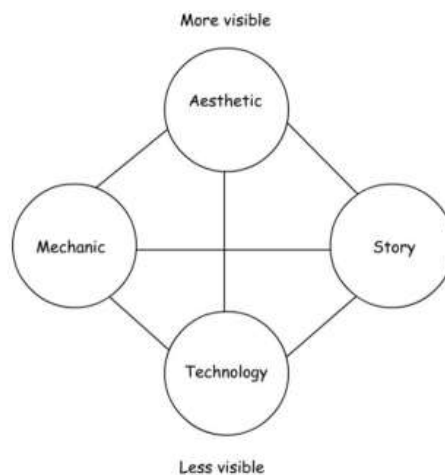
Game adalah objek yang terdiri dari komponen - komponen dan aturan - aturan, dengan kriteria tertentu (Kramer, 2006). *Game* juga berarti bentuk partisipatif atau interaktif, hiburan, dan berlangsung di dunia buatan yang diatur oleh peraturan. Mainan (*user*) dan *puzzle* tidak dapat dikatakan sebuah *game*. *Toys* adalah objek yang dimainkan tanpa peraturan. *Toys* dapat menjadi sebuah *game* jika terdapat beberapa peraturan untuk memainkannya. Akan tetapi biasanya sebuah *toy* muncul tanpa ada peraturan. Tidak seperti *toy*, *puzzle* memiliki satu aturan pasti dan memiliki solusi yang tepat yang sedang dicoba untuk ditemukan (Rollings *et al.*, 2003 : 34).

Game komputer semula adalah sebuah citra yang sederhana, dalam perkembangannya *game* berubah menjadi citra 2D yang kompleks bahkan menggunakan citra 3D (Putra *et al.*, 2013: 102). Salah satu contoh *game* yang cukup populer adalah *Metal Slug 2* untuk citra 2D dan *Grand Theft Auto* untuk citra 3D.

2.1.1 Elemen Game

Ada banyak cara untuk membuat klasifikasi elemen-elemen yang menyusun terbentuknya sebuah *game*. Ada beberapa kategori yang sudah tersusun sebagai elemen dari sebuah *game* bisa disebut dengan *elemental tetrad* (Schell

2008 : 41-42). Elemen *game* terdiri atas mekanisme (*mechanics*), cerita (*story*), estetika (*aesthetics*) dan teknologi (*technology*). *Elemental tetrad* ditunjukkan pada gambar 2.1.



Gambar 2.1 Elemen Dasar *Game*

(Schell , 2008: 42)

1. Mekanisme (*Mechanics*)

Mekanisme adalah prosedur dan aturan dalam sebuah *game* atau bisa disebut dengan sistem dari sebuah *game*. Mekanisme akan mendeskripsikan tujuan dari sebuah *game*, bagaimana pemain dapat melakukan sebuah aksi, termasuk runtutan kejadian yang akan terjadi apabila pemain mencoba melakukan sebuah aksi. Berbeda dengan hiburan lain yang lebih linier seperti buku, film, dan sebagainya, *game* akan lebih melibatkan teknologi, cerita, dan estetika, dan sebuah mekanisme.

Mekanisme yang terdapat dalam sebuah *game* membuat *game* berbeda dari hiburan lain. Dalam membangun sebuah mekanisme atau *gameplay* dalam

sebuah *game* perlu adanya keselarasan mekanisme dengan skenario cerita yang ada di dalam *game*.

2. Cerita (*Story*)

Cerita adalah serangkaian dari *events* yang mendukung sebuah permainan. Skenario cerita bisa berwujud sebuah cerita linier yang sudah terdapat pada *script*, maupun cerita yang bersifat bercabang dan dapat ditentukan oleh pemain. Ketika sebuah terdapat cerita untuk disampaikan dalam *game*, harus memilih mekanisme yang tepat dan selaras sehingga dapat memberikan kesan dari sebuah cerita dapat diperkuat.

3. Estetika (*Aesthetics*)

Estetika adalah unsur seni dalam sebuah *game*. Estetika berperan agar *game* terlihat, terdengar, dan terasa menarik. Estetika adalah sebuah aspek penting dalam sebuah desain *game*, dikarenakan estetika berkaitan langsung dengan pengalaman pemain dalam memainkan *game*. Ketika seorang pemain merasakan nada atau tampilan tertentu maka pemain bisa terbawa dalam suasana *game*. Pemilihan estetika yang tepat dapat membuat pemain tidak mudah melupakan pengalamannya dalam memainkan sebuah *game*.

4. Teknologi (*Technology*)

Teknologi yang dimaksud dalam elemen *game* adalah material dan interaksi yang membuat sebuah *game*, seperti plastik, kertas dan pensil. Teknologi yang dipilih untuk sebuah *game*, memungkinkan *game* memiliki *rule* (peraturan) untuk melakukan sesuatu dan mencegah melakukan sesuatu

yang lain. Dengan kata lain teknologi adalah sebuah perantara antara estetika, mekanisme, dan cerita.

2.1.2 Genre *Game*

Genre *game* merupakan elemen kunci desain untuk menentukan tahap pembuatan, *backstory* dan dialog *game* (Meigs 2003: 108). Genre *game* yang berkembang pada masa kini memiliki banyak cabang, diantaranya adalah *real time strategy*, *role play game*, *first person shooter*, dan masih banyak lagi (Crawford :2003). Beberapa *genre game* yang sering dikembangkan (Rollings *et al.*, 2003: 289) antara lain *action games*, *strategy games*, *adventures games*, *sports games*, dan *artificial life*.

1. *Action Games*

Action games adalah jenis *game* yang paling lama di dunia. Contoh *Action Games* yang terkenal adalah Mario Bross, Spiderman, dan lain-lain. Untuk mengenali *Action game*, terdapat ciri-ciri khas antara lain:

- a. Berorientasi kecepatan dan refleks,
- b. *Gameplay* yang sederhana, serta
- c. Cocok bagi *casual gamers*.

2. *Strategy Games*

Strategy games bermula dari sebuah *board games* sebelum adanya *game* komputer. Contoh *Strategy Games* adalah: *WarCraft*, *DOTA 2*, *Rise Of Nation*, *Age Of Empires*, dan lain-lain. Ciri-ciri dari *strategy games* antara lain:

- a. Berorientasi kepada logika dan sumber daya;
- b. Komputer memungkinkan manusia untuk bisa mengatur sumberdaya;
- c. Imajinatif, artinya apa yang ada dalam sebuah *game* yang tidak mungkin bisa dilakukan di kehidupan nyata; dan
- d. Cocok bagi *gamer* yang menyukai pemikiran dan pengaturan sumber daya.

3. *Adventures Games*

Adventure Games berorientasi sebagai cerita interaktif dari karakter utama yang dikontrol oleh pemain. Contoh *game* ini adalah *So Blonde*, *Myst*, *Indiana Jones*, dan lain-lain.

4. *Sports Games*

Sports games berusaha meniru kehidupan nyata. Olahraga yang ada di dunia nyata disimulasikan secara virtual dalam *sport games*. *Sport Games* yang terkenal diantaranya FIFA, PES (*Pro Evolution Soccer*), dan lain-lain. Ciri-ciri *game* jenis ini:

- a. Peraturan yang ada di olahraga nyata dipakai sebagai *rule* (peraturan) dalam *game*;
- b. Berorientasi kompetisi antar pemain; serta
- c. Lebih menyenangkan bila dimainkan bersama.

5. *Artificial Life*

Artificial Life Games memberikan gambaran kehidupan nyata. Di dalam *game* diciptakan sebuah dunia virtual yang menyerupai asli beserta kehidupannya. Contoh *game* ini adalah *The Sims*, *Sims Pets*, dan lain-lain.

2.1.3 Word Puzzle Game

Pada kehidupan kita sekarang ini, keberadaan *game* atau yang kita kenal dengan istilah permainan sangatlah penting. Permainan bukan hanya sebagai pengisi waktu senggang, tetapi juga sebagai sarana hiburan bagi sebagian besar dari kita yang memiliki banyak kesibukan. Salah satu permainan yang banyak digemari saat ini adalah *game* komputer, diantaranya adalah permainan “*word puzzle game*”. *Word puzzle game* adalah *game* yang berupa teka-teki yang dimainkan dengan cara menjawab teka-teki yang telah diberikan dan menuliskan jawabannya pada *field* yang telah disediakan. Sebagian besar orang memainkan permainan ini dengan hanya mengandalkan penglihatan dan imajinasi untuk menemukan kata-kata atau jawaban yang tersembunyi pada permainan. Dalam skripsi ini, penulis akan memaparkan analisa penulis terhadap penerapan algoritma *string matching* pada *word puzzle game*.

2.2 Android

2.2.1 Sistem Operasi Android

Menurut Meier (2009), Android adalah sebuah *software stack* bersifat *open source* yang mencakup sistem operasi, *middleware*, dan *key applications* beserta sekumpulan *Application Programming Interface* (API) untuk merancang sebuah aplikasi *mobile* dengan menggunakan bahasa pemrograman Java. Aplikasi berbasis android dapat diciptakan, dikembangkan secara bebas serta dapat dengan mudah diunduh dan digunakan sesuai kebutuhan pengguna.

Android merupakan sebuah sistem operasi perangkat *mobile* seperti *smartphone*, *tablet*, *laptop* bahkan televisi dan perangkat permainan berbasis linux. Android menyediakan kebebasan bagi para pengembang aplikasi untuk menciptakan aplikasi mereka sendiri. Android adalah sebuah *platform open source* yang memungkinkan untuk digunakan dalam penyelesaian masalah yang berkaitan dengan bantuan aplikasi khusus, pengolahan data yang terhubung dengan basis data di *server*, media permainan, komunikasi, dan lain sebagainya.

Dalam bukunya “Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android”, H., Nazruddin Safaat (2012) mengungkapkan bahwa android adalah *mobile platform* pertama yang lengkap, terbuka, dan bebas. Hal ini diuraikan sebagai berikut:

1. Lengkap (*complete platform*)

Para desainer aplikasi dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi

2. Terbuka (*open source platform*)

Platform Android disediakan melalui lisensi *open-source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi.

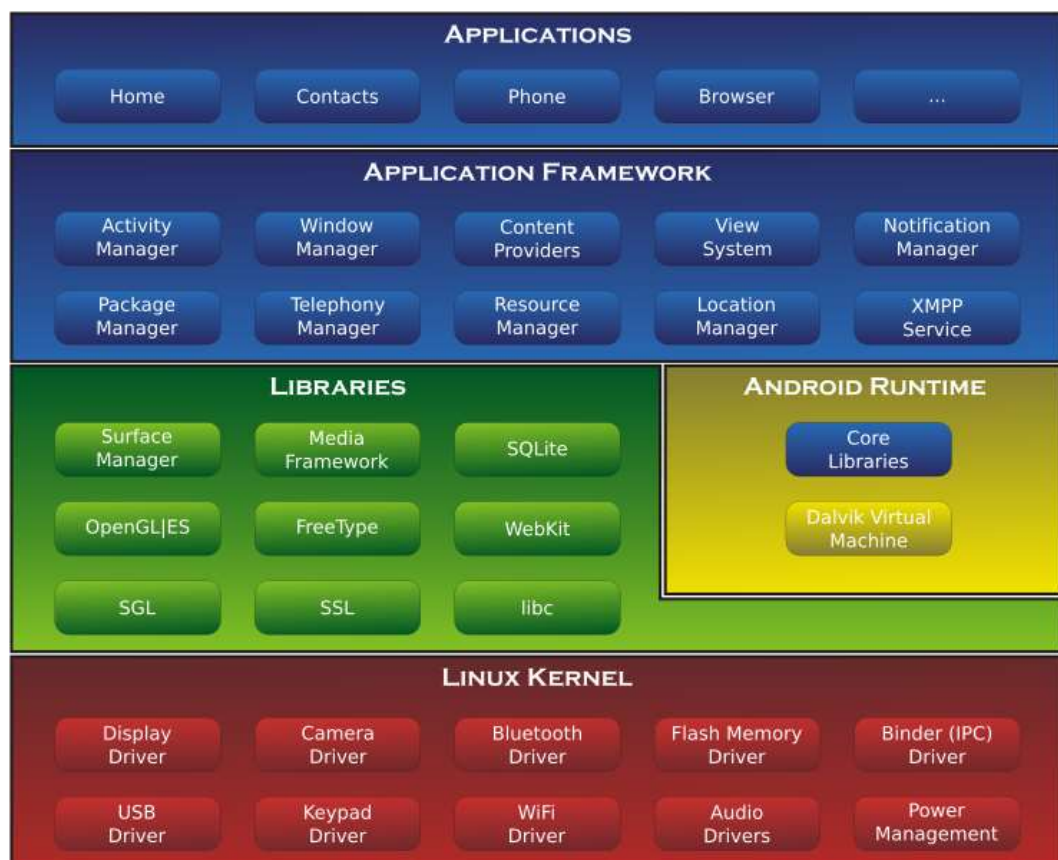
3. Bebas (*free platform*)

Android adalah platform yang bebas untuk para *developer* aplikasi untuk membuat aplikasi mereka sendiri. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan.

Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.2.2 Arsitektur Android

Secara garis besar arsitektur Android dapat digambarkan sebagai beberapa lapisan/*layer* yang saling terhubung. Pada dasarnya arsitektur Android ada 5 lapisan, yaitu *Applications* dan *Widget*, *Applications Framework*, *Libraries*, *Android Runtime*, *Linux Kernel*. Hal ini ditunjukkan pada gambar 2.2.



Gambar 2.2 Arsitektur Android

1. *Applications dan Widget*

Application dan *widget* adalah *layer* yang berhubungan dengan aplikasi saja, di mana biasanya aplikasi yang sudah ter-*download* akan di instalasi lalu dijalankan. Di *layer* ini terdapat aplikasi inti termasuk aplikasi klien email, aplikasi SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis dengan bahasa pemrograman java.

2. *Applications Framework*

Android adalah sebuah “*Open Development Platform*” yang menawarkan kepada pengembang untuk mengakses penuh *API framework* seperti yang dilakukan di *layer/lapisan* ini. Arsitektur aplikasi dirancang dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*). Dapat disimpulkan *layer/lapisan* ini adalah lapisan di mana para pengembang aplikasi melakukan pengembangan/ pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada lapisan inilah aplikasi dirancang dan dibuat.

3. *Libraries*

Libraries adalah lapisan di mana fitur-fitur Android berada, biasanya para pengembang aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4. *Android Runtime*

Android runtime adalah *layer* yang membuat aplikasi Android dapat dijalankan di mana di dalam prosesnya menggunakan implementasi linux.

Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android.

5. *Linux Kernel*

Linux kernel adalah lapisan/*layer* di mana inti dari sistem operasi Android itu berada. Berisi file-file *system* yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan lain sebagainya.

2.3 Algoritma *String Matching*

Algoritma *string matching* dalam bahasa Indonesia dikenal dengan istilah algoritma pencocokan *string* (Munir, 2007). Persoalan pencarian *string* dirumuskan sebagai berikut:

Diberikan:

1. Sebuah teks (*text*), yaitu sebuah (*long*) *string* yang panjangnya n karakter.
2. *Pattern*, yaitu sebuah *string* dengan panjang m karakter ($m < n$) yang akan dicari dalam teks.

Carilah lokasi pertama di dalam teks yang bersesuaian dengan *pattern*. Aplikasi permasalahan pencocokan *string* biasa ditemukan dalam pencarian sebuah kata dalam dokumen (misalnya menu *Find* dalam Microsoft Word).

Contoh :

```

Pattern : not
Teks    : nobody noticed him
                ↑target

```

Dalam algoritma pencocokan *string*, teks diasumsikan berada di dalam memori, sehingga bila kita mencari *string* di dalam sebuah arsip, maka semua isi

arsip perlu dibaca terlebih dahulu kemudian disimpan di dalam memori. Jika *pattern* muncul lebih dari sekali di dalam teks, maka pencarian hanya akan memberikan keluaran berupa lokasi *pattern* ditemukan pertama kali. Ada tiga buah algoritma yang umum digunakan dalam melakukan pencocokan *string*, yaitu algoritma *Brute Force*, *Knuth Morris Pratt*, serta *Boyer Moore*.

2.4 Algoritma Knuth Morris Pratt

Pada algoritma *Knuth-Morris-Pratt* (KMP) informasi ketidakcocokan *pattern* dengan teks digunakan disimpan untuk menentukan jumlah pergeseran. Algoritma KMP melakukan pergeseran lebih jauh sesuai dengan informasi yang disimpan, tidak seperti pada algoritma *Brute Force* di mana pergeseran dilakukan setiap satu karakter, sehingga waktu pencarian dapat dikurangi secara signifikan.

Algoritma *Knuth-Morris-Pratt* dikembangkan oleh D. E. Knuth, bersama-sama dengan J.H Morris dan V. R. Pratt (Munir, 2007). Algoritma KMP ditunjukkan sebagai berikut.

Misalkan A adalah alfabet dan $x = x_1x_2\dots x_k$, dengan $k \in \text{himpunan } \mathbb{N}$, adalah *string* yang panjangnya k yang dibentuk dari karakter-karakter di dalam alfabet A . Awalan (*prefix*) dari x adalah *up string (substring)* u , dengan $u = x_1x_2\dots x_{k-1}$, $k \in \{1, 2, \dots, k-1\}$. Dengan kata lain, x diawali dengan u . Akhiran (*suffix*) dari x adalah *up string(substring)* u , dengan $u = x_{k-b}x_{k-b+1}\dots x_k$, $k \in \{1, 2, \dots, k-1\}$. Dengan kata lain, x diakhiri dengan v . Pinggiran (*border*) dari x adalah *up string* r sedemikian sehingga $r = x_1x_2\dots x_{k-1}$ dan $u = x_{k-b}x_{k-b+1}\dots x_k$, $k \in \{1, 2, \dots, k-1\}$. Dengan b

adalah fungsi pinggiran. Dengan kata lain, pinggiran dari x adalah *upastring* yang keduanya awalan dan juga akhiran sebenarnya dari x .

Algoritma KMP melakukan proses awal terhadap *pattern* P dengan menghitung fungsi pinggiran yang mengindikasikan pergeseran s terbesar yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum pencarian *string* (Munir, 2007). Fungsi pinggiran hanya bergantung pada karakter-karakter di dalam *pattern*. Fungsi pinggiran $b(j)$ didefinisikan sebagai ukuran awal terpanjang dari P yang merupakan akhiran dari $P[1..j]$.

2.5 Interactive Multimedia System Design and Development

Metode *Interactive Multimedia System Design and Development* (IMSDD) adalah suatu metode perancangan dan pengembangan sistem aplikasi multimedia interaktif termasuk *game* yang terdiri atas tahapan-tahapan yang terstruktur. IMSDD dapat dibagi menjadi beberapa tahap, yaitu *System Requirements*, *Design Considerations*, *Implementation*, dan *Evaluation* (Dastbaz 2003: 130-132). Siklus IMSDD dapat ditunjukkan pada gambar 2.3.

1. System Requirements

Pada tahap ini *developer* akan mengumpulkan dan mempelajari data-data yang dibutuhkan oleh sistem. Fungsi-fungsi pada tahapan ini antara lain :

- a. Menentukan maksud dan tujuan dari sistem yang akan dibuat,
- b. Menentukan *user* dari sistem yang akan dibuat,
- c. Menentukan kebutuhan-kebutuhan yang akan digunakan seperti *hardware*, *software*, dan *authoring tools*, dan
- d. Mempertimbangkan *delivery platform*.

2. *Design Considerations*

Tahap ini berfokus pada bagaimana sistem berfungsi sebagaimana yang diharapkan kemudian disempurnakan oleh *developer* dengan memberikan deskripsi yang cukup mendetail pada sistem. Permasalahan yang dibahas pada tahapan ini adalah:

- a. Menentukan media yang akan digunakan untuk *design interface* pada sistem (*design metaphor*),
- b. Menentukan tipe informasi yang diperlukan untuk diintegrasikan ke sistem yang akan dibuat seperti teks, gambar, suara, video, dan animasi (*information types and formats*),
- c. Menunjukkan hubungan struktur dan fitur-fitur untuk menghindari masalah (*navigational structures*),
- d. Menjelaskan tipe-tipe dan fitur-fitur dari *tools* yang diperlukan oleh sistem (*system control*).

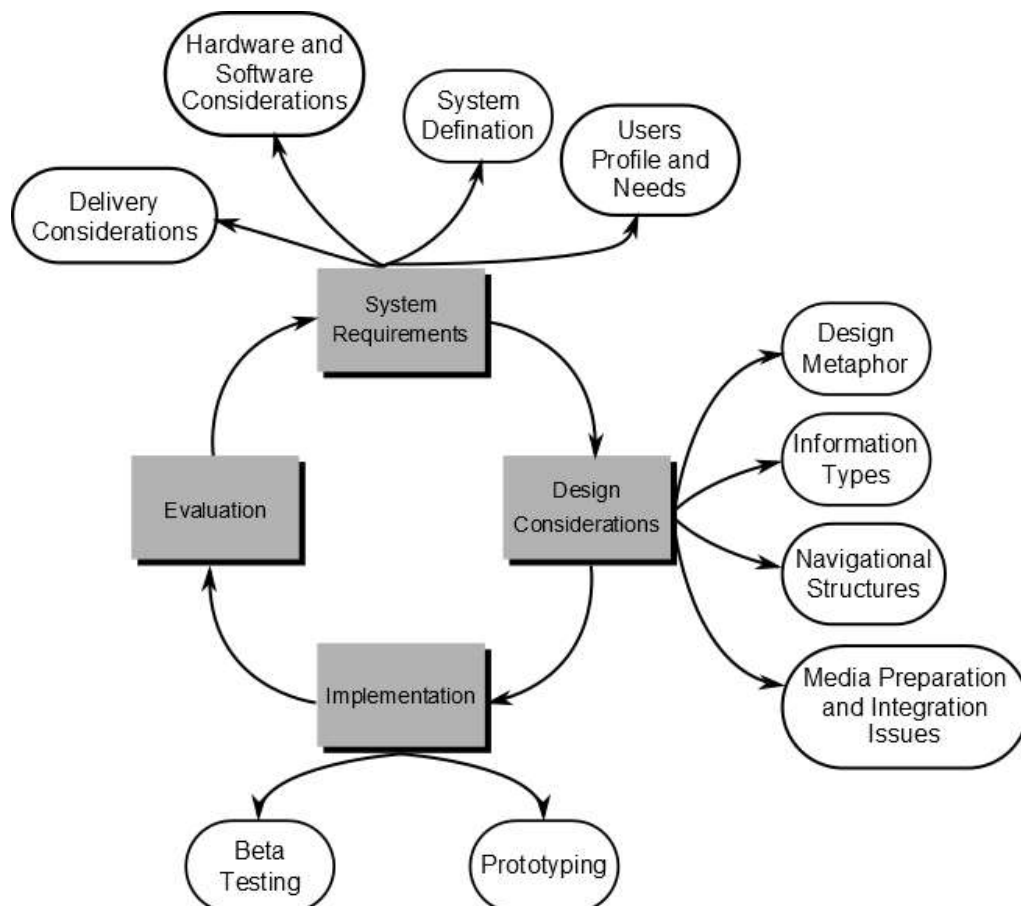
3. *Implementation*

Dalam tahap implementasi program, komponen-komponen dari program akan diuji untuk mengetahui apakah program yang bersangkutan dapat berjalan dengan baik berdasarkan kriteria-kriteria yang telah ditetapkan.

Maka dari itu perlu melakukan beberapa hal, yaitu membuat *prototype* dan melakukan *beta test*.

4. Evaluation

Evaluasi adalah tahap dimana sistem dievaluasi terhadap tujuan awal dibentuknya sebuah program.



Gambar 2.3 Siklus IMSDD

(Dastbaz, 2003:131)

2.6 Unified Modeling Language (UML)

UML merupakan sebuah *blue print* dari suatancangan sistem (Whitten *et al.*, 2007:381). Sedangkan UML sendiri telah menjadi standar penulisan *blue print* sebuah perangkat lunak (Fowler, 1996: 13). Dalam UML ada beberapa diagram yang menggambarkan rancangan *software* yang dibahasakan dalam gambar yang dapat dipahami oleh manusia, diantaranya adalah *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

1. Use Case Diagram

Use case diagram adalah diagram yang menunjukkan suatu set penggunaan kasus beserta aktornya (suatu kelas spesial), beserta hubungannya. *Use case diagram* mengamatkan suatu pandangan kasus statis pada suatu sistem. Fungsi utama dari *use case* adalah untuk menyusun dan membuat model tingkah laku (*behaviour*) dari suatu sistem.

2. Activity Diagram

Activity diagram menunjukkan alur atau langkah dari aktifitas ke aktifitas lain dalam yang mengamatkan secara dinamis suatu sistem. Diagram ini sangat penting dalam membuat model fungsi dari suatu sistem dan menekankan pada alur kontrol diantara objek-objek yang bersangkutan. Pada setiap *use case* yang ada, setidaknya terdapat satu *activity diagram*.

3. Sequence Diagram

Sequence diagram adalah suatu diagram yang bagaimana objek – objek yang terlibat di dalam suatu *use case* melakukan interaksi dan komunikasi dari objek yang ada. Diagram ini juga menunjukkan serangkaian

pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu.

4. *Class Diagram*

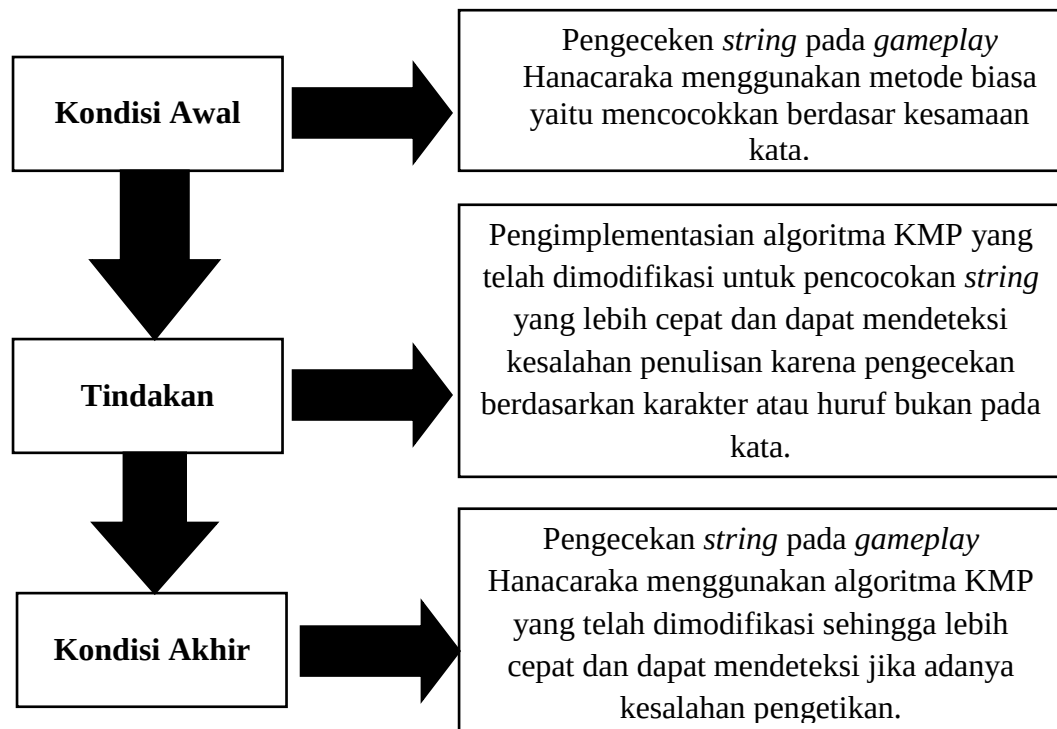
Class diagram menunjukkan suatu set kelas serta hubungannya antar *class*. Diagram ini adalah diagram yang umum ditemukan pada sistem berbasis objek. *Class diagram* mengamati desain yang statis dari suatu sistem.

2.7 Kerangka Berfikir

Sugiyono (2010) mencoba menjelaskan tentang kerangka berpikir dengan mengutip dari Uma Sekaran (1992) dengan menuliskan bahwa kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting.

Aksara Jawa merupakan salah satu peninggalan budaya yang tak ternilai harganya. Bentuk aksara dan seni pembuatannya pun menjadi peninggalan yang patut untuk dilestarikan. Aksara ini menjadi bukti nyata adanya zaman terdahulu sebelum adanya bangsa Indonesia. Salah satu upaya melestarikannya adalah dengan melakukan digitalisasi Aksara Jawa, sehingga bangsa Indonesia tidak akan kehilangan nilai budayanya.

Perkembangan teknologi terutama teknologi *game* merupakan salah satu teknologi yang dapat dimanfaatkan untuk melestarikan Aksara Jawa ini. Selain itu dapat juga dimanfaatkan untuk menjadi media pembelajaran Aksara Jawa. Adapun kerangka berfikir dalam penelitian ini dapat dilihat pada gambar 2.4.



Gambar 2.4 Kerangka Berfikir

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Pelaksanaan

Penelitian ini dilaksanakan pada bulan November – Desember 2014 di Laboratorium Teknik Elektro, Fakultas Teknik Universitas Negeri Semarang.

3.2 Jenis Penelitian dan Desain Penelitian

Penelitian ini menggunakan metode *Interactive Multimedia System Design and Development* (IMSDD) yang digunakan untuk merancang perangkat lunak *game* Hanacaraka. Metode ini merupakan pengembangan dari metode *waterfall* yang diadaptasi sesuai dengan kebutuhan pengembangan aplikasi multimedia, sehingga metode ini merupakan metode yang paling cocok dalam pengembangan *game* Hacaraka.

3.2.1 System Requirements

Sebelum mengembangkan produk ini, perlu adanya sebuah analisis untuk mengetahui kebutuhan-kebutuhan yang diperlukan dalam membangun sebuah *game*. Analisis kebutuhan-kebutuhan tersebut meliputi analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, dan perangkat lunak yang digunakan pengguna.

1. Analisis Kebutuhan Perangkat Keras

Dalam proses pembuatan *game* Hanacaraka dibutuhkan komputer/ *notebook* untuk proses *scripting*. Sedangkan pengujian aplikasi menggunakan *smartphone* dengan sistem operasi Android. *Notebook* yang digunakan adalah Toshiba Satellite C640, spesifikasi dapat dilihat pada tabel 3.1. *Smartphone* yang digunakan untuk pengujian adalah SONY Xperia S (LT26i), spesifikasi dapat dilihat pada tabel 3.2.

Tabel 3.1 Spesifikasi Toshiba Satellite C640

Prosesor	Intel® Pentium™ P6000 1,9 Ghz Processor
Memori	DDR3 1600 MHz SDRAM, 3 GB
Display	14.0" 16:9 HD (1366x768) LED Backlight
Grafis	Integrated Intel® HD Graphics 3000
Storage	160GB HDD
Drive Optik	Super-Multi DVD
Card Reader	Card Reader (SD)
Kamera	HD Web Camera
Networking	Terintegrasi 802.11 a/b/g/n Built-in Bluetooth™ V4.0 (Optional) 10/100/1000 Base T
Interface	1 x COMBO audio jack 1 x VGA port/Mini D-sub 15-pin for external monitor 2 x USB 2.0 port(s) 1 x RJ45 LAN Jack for LAN insert 1 x HDMI
Audio	Built-in Speakers And Microphone

Tabel 3.2 Spesifikasi SONY Xperia S (LT26i)

Prosesor	1,5 GHz Qualcomm Dual Core
Chipset	Qualcomm MSM8260 Snapdragon
CPU	Dual-core 1.5 GHz
GPU	Adreno 220
Memori	32 GB, 1 GB RAM
Display	LED-backlit LCD, capacitive touchscreen, 16M colors 720 x 1280 pixels, 4.3 inches (~342 ppi pixel density)
Camera	12 MP, 4000 x 3000 pixels, autofocus, LED flash
OS	Android OS, v2.3 (Gingerbread), upgradable to v4.1.2 (Jelly Bean)

2. Analisis Kebutuhan Perangkat Lunak

Sebuah *game* yang sesuai dengan kebutuhan dari *player* yang akan memainkan akan dibangun. Tuntutan *player* adalah adanya sebuah *game* yang memiliki kriteria sebagai berikut:

- a. Mengesankan dan menarik
- b. Tidak membosankan
- c. Interaktif
- d. Visualisasi dan sistem yang baik serta permainan menantang

Game yang dibuat berjudul “Hanacaraka” menggunakan bahasa pemrograman C# dengan bantuan tools Unity3D.

3. Perangkat Lunak yang Digunakan Pengembang

Perangkat lunak yang digunakan pengembang pada penelitian ini adalah sebagai berikut:

- a. Unity3D
- b. Visual Studio 2010
- c. Android SDK
- d. Java Development Kit (JDK)
- e. Inkscape
- f. Audacity
- g. Bfxr

3.2.2 Design Consideration

Pada bagian ini akan dibahas mengenai analisis sistem dan perancangan sistem pada *game* Hanacaraka. Pada analisis sistem ini akan dilakukan analisis terhadap *game* Hanacaraka dan algoritma *Knuth Morris Pratt*.

3.2.2.1 Analisis Permainan

Analisis dan deskripsi dari permainan Hanacaraka yang akan dirancang meliputi informasi umum *game* Hanacaraka, tujuan pembuatan *game* Hanacaraka, konsep dasar *game* Hanacaraka, komponen permainan *game* Hanacaraka, *use case game* Hanacaraka, dan *activity diagram game* Hanacaraka.

1. Informasi Umum *Game* Hanacaraka

Hanacaraka adalah *game* ber-genre *word puzzle game* yang dapat digunakan sebagai media pengenalan aksara Jawa bagi orang umum dengan implementasi algoritma *Knuth Morris Pratt* pada pencocokan *string*-nya yang dapat dimainkan pada *platform mobile* berbasis OS *Android*.

Dalam *game* ini, terdapat soal yang ditulis dengan aksara jawa dan juga gambar sebagai pendukung, kemudian pemain diminta untuk menerjemahkan aksara jawa tersebut ke dalam aksara latin kemudian digabungkan dengan gambar yang ada untuk mengetahui jawaban dari soal tersebut. Setelah itu menuliskan jawabannya ke dalam *field* yang sudah disediakan.

2. Tujuan Pembuatan *Game* Hanacaraka

Game ini dibuat untuk memberikan variasi dan gaya berbeda dalam media pengenalan aksara jawa kepada masyarakat umum. Di dalam *game* ini selain sebagai media pengenalan aksara jawa, juga terkandung unsur asah logika yang terdapat pada penggabungan aksara jawa dan gambar untuk kemudian mencari jawaban dari soal yang akan membuat orang semakin tertantang untuk memainkannya dan semakin bersemangat untuk mengenal dan mempelajari aksara jawa.

3. Konsep Dasar *Game* Hanacaraka

Dalam *game* Hanacaraka, *player* akan diminta untuk menjawab soal-soal yang berisi aksara jawa yang dipadukan dengan gambar pendukung untuk kemudian *player* menginputkan jawaban dari soal tersebut ke dalam *field* yang telah tersedia. Ada beberapa tipe soal yang tersedia. Dari mulai tipe soal tentang aksara carakan tanpa sandangan, sampai angka jawa. Dalam *game* Hanacaraka pemain dibekali dengan lima nyawa yang akan berkurang jika si pemain salah dalam menjawab soal. Apabila nyawa yang diberikan telah habis, maka permainan akan selesai dan *game* akan dimulai dari awal lagi.

Selain nyawa, saat pertama kali memainkan *game* Hanacaraka, pemain akan diberi lima buah bantuan untuk membantu menjawab soal yang sulit untuk dijawab. Dan jumlahnya akan berkurang seiring dengan jumlah bantuan yang digunakan oleh pemain.

Dalam *game* Hanacaraka, terdapat enam level dengan tingkat kesulitan dan tipe soal yang berbeda beda. Setiap level terdapat lima soal yang harus diselesaikan. Dan dari masing-masing soal terdapat tiga variasi soal yang akan di *load* secara random, kecuali pada soal ke lima dari tiap level hanya terdapat satu variasi soal. Sehingga jika dijumlahkan secara keseluruhan maka terdapat 78 variasi soal pada *game* Hanacaraka.

Secara *default*, saat pertama kali memainkan *game* Hanacaraka hanya terdapat satu level yang bisa dimainkan, sedangkan level lainnya masih dikunci. Untuk bisa membuka level selanjutnya, maka pemain harus menyelesaikan minimal tiga soal dari level yang telah terbuka. Soal ke lima dari masing-masing level merupakan soal raja dan paling sulit untuk di selesaikan di level tersebut. Apabila pemain telah mengalahkan soal raja di masing-masing level, maka pemain akan mendapatkan medali untuk tiap level yang dimainkan.

4. Komponen Permainan *Game* Hanacaraka

Dalam *game* Hanacaraka, terdapat beberapa komponen permainan, antara lain soal, *lock* level dan *lock* soal, medali level, bantuan, dan nyawa.

a. Soal

Soal yang terdapat pada *game* Hanacaraka terdapat dua kategori, yaitu soal penyisihan dan soal raja. Soal penyisihan merupakan soal nomor satu sampai dengan nomor empat di masing-masing level. Sedangkan soal raja adalah soal nomor lima di masing-masing level.

b. *Lock* level dan *lock* soal

Lock soal merupakan kunci yang menghalangi pemain untuk pergi ke soal selanjutnya sebelum menyelesaikan soal terakhir yang terdapat di masing-masing level. Sedangkan *lock* level merupakan kunci yang menghalangi pemain untuk pergi ke level selanjutnya sebelum menyelesaikan minimal tiga soal dari level yang sedang dimainkan.

c. Medali level

Medali level merupakan *reward* yang diberikan kepada pemain yang telah menyelesaikan semua soal di masing-masing level. Terdapat satu medali di masing-masing level, sehingga total ada lima medali yang bisa didapatkan.

d. Bantuan

Merupakan bantuan yang diberikan kepada pemain ketika si pemain menghadapi kesulitan untuk menjawab soal yang ada. Saat pertama kali bermain diberikan lima bantuan kepada pemain yang

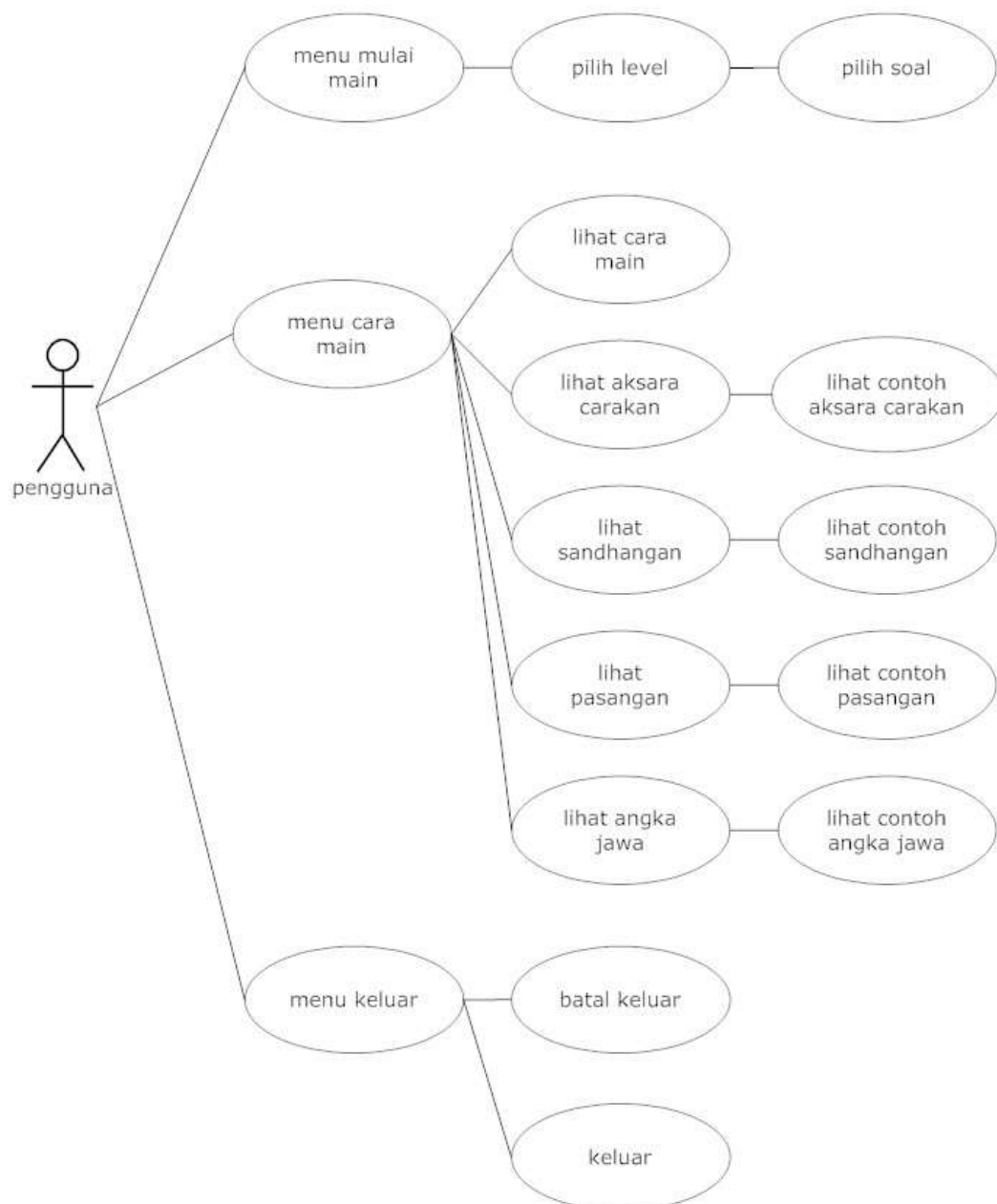
jumlahnya akan berkurang seiring dengan banyaknya jumlah bantuan yang digunakan oleh pemain.

e. Nyawa

Merupakan jumlah kesempatan yang diberikan kepada pemain untuk menjawab salah. Apabila nyawa yang diberikan sudah habis, maka permainan akan selesai atau *game over*. Pada saat pertama kali memulai permainan pemain dibekali dengan 5 nyawa yang akan berkurang sesuai dengan banyaknya jawaban salah yang diberikan oleh pemain.

5. *Use Case Game* Hanacaraka

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Pada tahap ini akan dibahas *use case diagram* program yang dilakukan *user* terhadap aplikasi yang sudah siap digunakan. *Use Case Diagram* tersebut akan dijelaskan pada gambar 3.1.



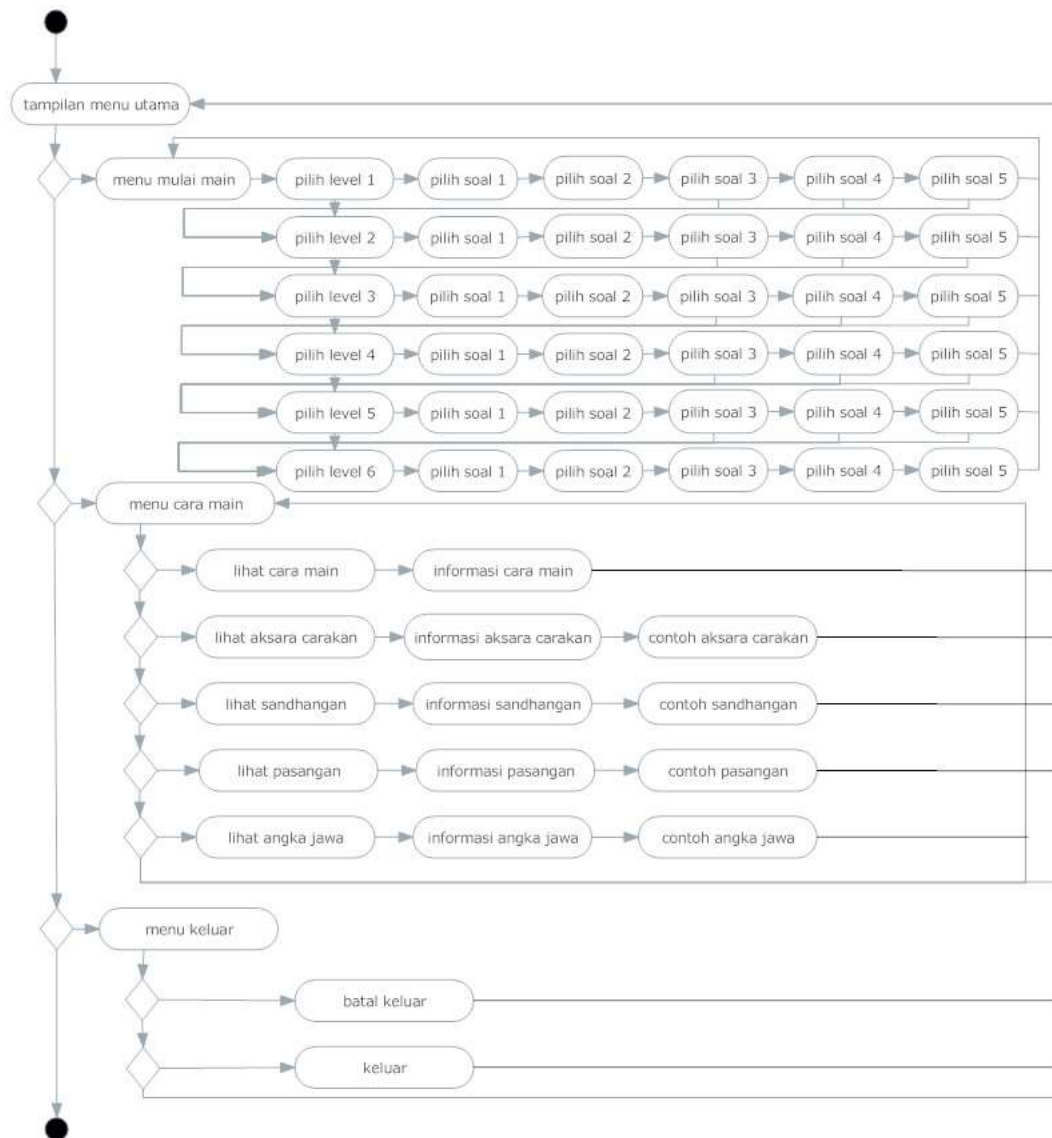
Gambar 3.1 Use Case Diagram Hanacaraka

Pada diagram yang ditunjukkan pada gambar 3.1 tersebut memiliki 1 *actor* yaitu Pengguna, dan memiliki 16 *Use Case*. Dimana pengguna dapat menjalankan sistem berupa melihat menu mulai main, cara main, dan menu keluar. Saat pengguna memilih menu mulai main, maka akan ada pilihan

menu pilih level, lalu pilih soal, maka akan menampilkan soal yang harus diselesaikan. Jika pengguna memilih menu cara main, maka pengguna akan bisa melihat cara main yang berisi tentang cara main, lalu aksara carakan, sandhangan, pasangan, dan angka jawa yang masing-masing disertai contoh. Kemudian apabila pemain memilih menu keluar, maka pemain akan bisa memilih apakah ingin keluar atau batal untuk keluar.

6. *Activity Diagram Game Hanacaraka*

Activity Diagram menggambarkan alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Pada tahapan ini akan dibahas, *activity diagram* program yang dilakukan *user* terhadap aplikasi yang sudah siap digunakan. *Activity diagram* tersebut akan dijelaskan melalui gambar 3.2.



Gambar 3.2 Activity Diagram

Pada diagram yang ditunjukkan pada gambar 3.2 tersebut menggunakan satu swimline yaitu Pengguna dan memiliki 49 activity. Dimana pengguna dapat memilih menu yang terdapat di aplikasi seperti menu mulai main, cara main, dan keluar. Didalam setiap menu utama terdapat beberapa cabang menu atau aktifitas yang dapat dilakukan. Misalnya apabila kita memilih menu cara main maka kita akan bisa

memilih untuk melihat cara main, aksara carakan, sandhangan, pasangan, serta melihat angka jawa. Di dalam sub menu tersebut juga terdapat sub menu yang lain, misalnya didalam sub menu aksara carakan kita bisa melihat menu contoh dari penggunaan aksara carakan.

3.2.2.2 Analisis Algoritma *Knuth Morris Pratt*

Algoritma yang digunakan untuk pencocokan *string* pada *game* Hanacaraka adalah menggunakan algoritma *Knuth Morris Pratt* atau sering disingkat menjadi algoritma KMP. Algoritma *Knuth Morris Pratt* sendiri adalah algoritma yang dikembangkan secara terpisah oleh D. E. Knuth pada tahun 1967, dan J.H Morris bersama V. R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977 (Munir, 2007).

Tidak seperti algoritma *Brute Force* yang mencocokkan *string* dengan melakukan pengecekan dan melakukan pergeseran setiap satu karakter, pada algoritma *Knuth Morris Pratt* informasi ketidak cocokan *pattern* dengan teks disimpan untuk menentukan jumlah pergeseran. Sehingga algoritma *Knuth Morris Pratt* melakukan pergeseran lebih jauh sesuai dengan informasi yang disimpan, yang menyebabkan waktu pencarian dapat dikurangi secara signifikan.

Selain itu, dari penyimpanan informasi pada pencocokan *string* yang dilakukan algoritma *Knuth Morris Pratt* dapat diambil keuntungan lain. Yaitu dari informasi yang disimpan dapat digunakan untuk menentukan berapa persen kedekatan *string* yang akan dicocokkan dengan *pattern* yang ada. Dan hal ini juga yang diimplementasikan pada *game* Hanacaraka. Sehingga apabila pemain salah

dalam memasukkan jawaban, maka akan dilihat berapa persen kesalahannya. Apabila kesalahan nya kurang dari 30 persen maka status jawabannya bukan “salah”, melainkan “hampir benar” dan nyawa yang diberikan kepada pemain tidak berkurang.

Pseudocode dari algoritma *Knuth Morris Pratt* pada fase pra-pencarian dan fase pencarian dapat dilihat pada listing 3.1 dan 3.2.

```

procedure preKMP(
    input P : array[0..n-1] of char,
    input n : integer,
    input/output kmpNext : array[0..n] of integer
)
Deklarasi:
i,j: integer

Algoritma
i := 0;
j := kmpNext[0] := -1;
while (i < n) {
    while (j > -1 and not(P[i] = P[j]))
        j := kmpNext[j];
    i := i+1;
    j := j+1;
    if (P[i] = P[j])
        kmpNext[i] := kmpNext[j];
    else
        kmpNext[i] := j;
    endif
endwhile

```

Listing 3.1 *Pseudocode* Algoritma KMP fase pra-pencarian

```

procedure KMPSearch(
    input m, n : integer,
    input P : array[0..n-1] of char,
    input T : array[0..m-1] of char,
    output ketemu : array[0..m-1] of boolean
)
Deklarasi:
I, j,next: integer
kmpNext : array[0..n] of integer

Algoritma:
preKMP(n, P, kmpNext)
i:=0
while (i<= m-n) do
    j:=0
    while (j < n and T[i+j] = P[j]) do

```

```

        j:=j+1
    endwhile
    if(j >= n) then
        ketemu[i]:=true;
    endif
    next:= j - kmpNext[j]
    i:= i+next
endwhile

```

Listing 3.2 *Pseudocode* Algoritma KMP fase pencarian

Berdasarkan *pseudocode* yang ditunjukkan pada listing 3.1 dan 3.2, dapat diketahui langkah-langkah yang dilakukan algoritma *Knuth Morris Pratt* pada saat mencocokkan string adalah sebagai berikut:

1. Algoritma *Knuth Morris Pratt* mulai mencocokkan *pattern* pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
 - b. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* berdasarkan tabel *next*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

Penerapan algoritma *Knuth Morris Pratt* pada permainan Hanacaraka dijelaskan dalam beberapa langkah berikut:

String S:

a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---

Pattern P:

c	a	r	a	k	a
---	---	---	---	---	---

Langkah 1: Bandingkan *pattern* [1] dengan *string* [1]

a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---



c	a	r	a	k	a
---	---	---	---	---	---

Pattern [1] tidak cocok dengan *string* [1]. Maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 2: Bandingkan *pattern* [1] dengan *string* [2]

a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---



c	a	r	a	k	a
---	---	---	---	---	---

Pattern [1] cocok dengan *string* [2]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak

akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [2] dengan *string* [3].

Langkah 3: Bandingkan *pattern* [2] dengan *string* [3]

a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---

↑

c	a	r	a	k	a
---	---	---	---	---	---

Pattern [2] cocok dengan *string* [3]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [3] dengan *string* [4].

Langkah 4: Bandingkan *pattern* [3] dengan *string* [4]

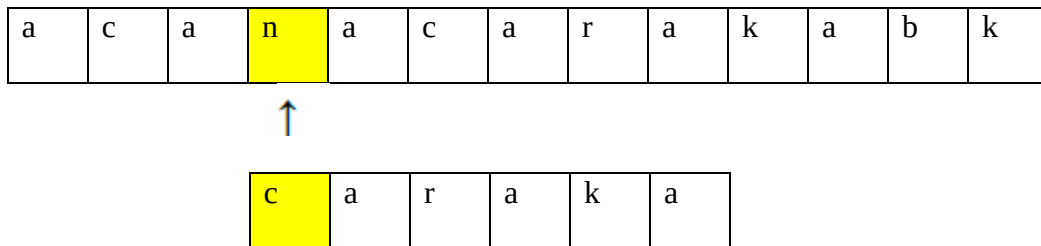
a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---

↑

c	a	r	a	k	a
---	---	---	---	---	---

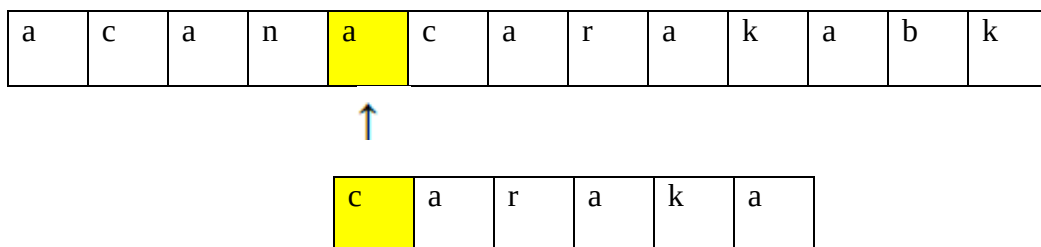
Pattern [3] tidak cocok dengan *string* [4]. Karena ada tidak ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menggunakan informasi yang telah disimpan pada langkah dua dan tiga untuk menentukan pergeseran selanjutnya. Sehingga pencocokan yang dilakukan bukanlah antara *pattern* [1] dengan *string* [3] seperti jika menggunakan algoritma *brute force*, namun antara *pattern* [1] dengan *string* [4] sehingga dapat mempercepat proses pencocokan *string*.

Langkah 5: Bandingkan *pattern* [1] dengan *string* [4]



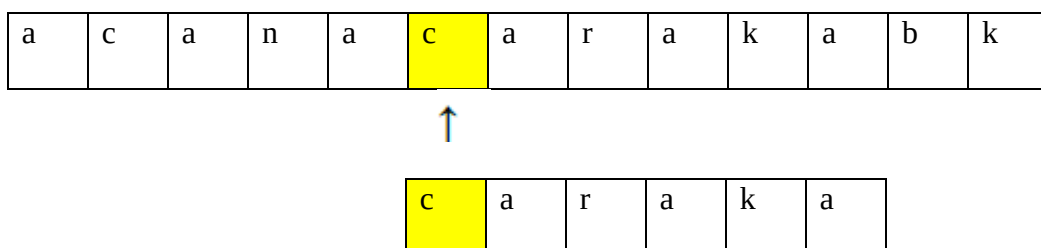
Pattern [1] tidak cocok dengan *string* [4]. Maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 6: Bandingkan *pattern* [1] dengan *string* [5]



Pattern [1] tidak cocok dengan *string* [5]. Maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 7: Bandingkan *pattern* [1] dengan *string* [6]



Pattern [1] cocok dengan *string* [6]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak

akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [2] dengan *string* [7].

Langkah 8: Bandingkan *pattern* [2] dengan *string* [7]

a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---

↑

c	a	r	a	k	a
---	---	---	---	---	---

Pattern [2] cocok dengan *string* [7]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [3] dengan *string* [8].

Langkah 9: Bandingkan *pattern* [3] dengan *string* [8]

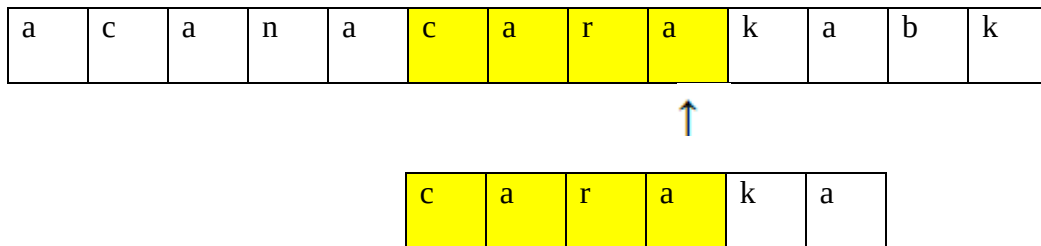
a	c	a	n	a	c	a	r	a	k	a	b	k
---	---	---	---	---	---	---	---	---	---	---	---	---

↑

c	a	r	a	k	a
---	---	---	---	---	---

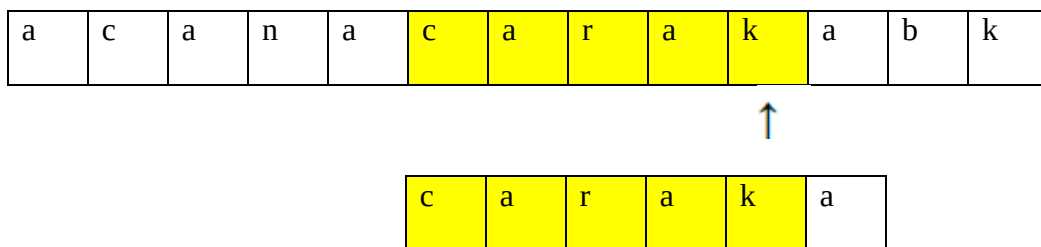
Pattern [3] cocok dengan *string* [8]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [4] dengan *string* [9].

Langkah 10: Bandingkan *pattern* [4] dengan *string* [9]



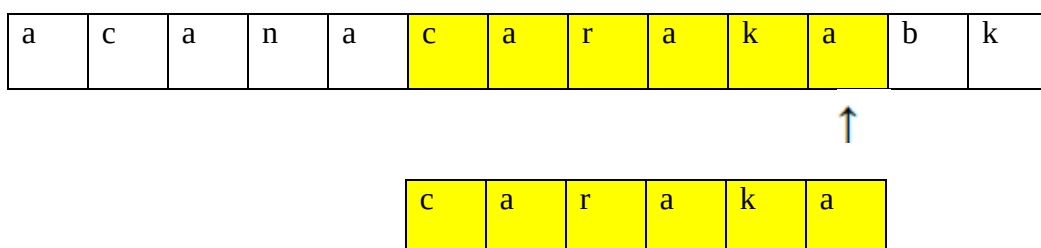
Pattern [4] cocok dengan *string* [9]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [5] dengan *string* [10].

Langkah 11: Bandingkan *pattern* [5] dengan *string* [10]



Pattern [5] cocok dengan *string* [10]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [6] dengan *string* [11].

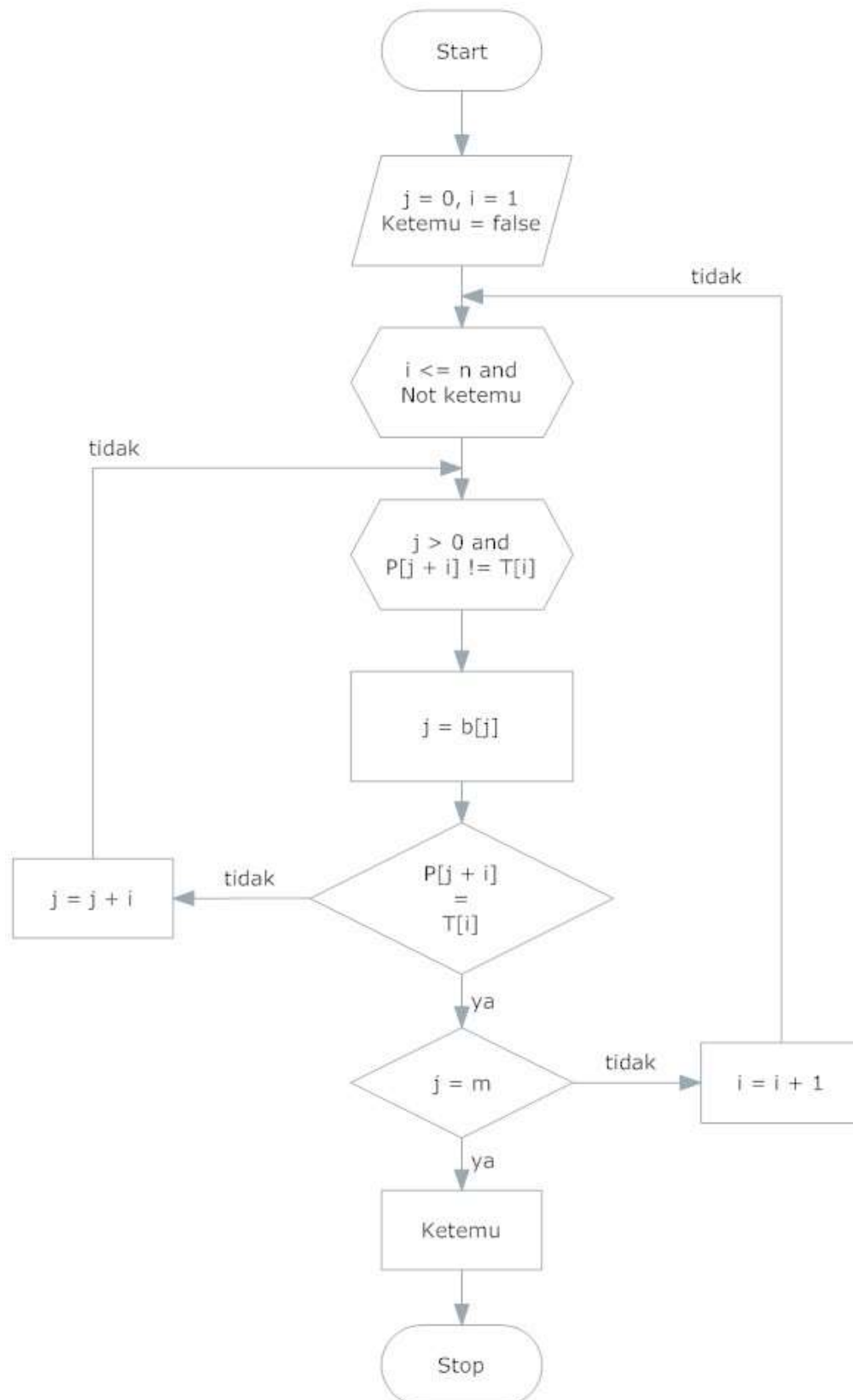
Langkah 12: Bandingkan *pattern* [6] dengan *string* [11]



Pattern [6] cocok dengan *string* [11]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [7] dengan *string* [12]. Namun karena jumlah *pattern* hanya 6 huruf, maka pencarian akan dihentikan dan diperoleh hasil bahwa *pattern* P terdapat kecocokan dengan *string* S sebesar 100 persen.

Jika semua huruf pada *pattern* sudah di cocokkan dengan *string* maka akan di temukan sebuah pola kosa kata di dalam *string*. Dalam menemukan sebuah pola *pattern* di dalam *string* akan dilakukan pergeseran beberapa kali untuk mencocokkan setiap huruf pada *pettern* yang di mulai dari sebelah kiri untuk mencocokkan setiap huruh pada *string*.

Contoh ilustrasi dari bagaimana algoritma *Knuth Morris Pratt* bekerja ini menggunakan *pattern* (caraka) yang panjangnya enam huruf melakukan pencocokan dengan *string* yang panjangnya tiga belas huruf acak untuk menemukan sebuah pola kata yang sama dengan *pattern* tersebut. Dengan melakukan pencocokan satu persatu di dalam *pattern* dengan seluruh huruh yang terdapat pada *string*. *Flowchart* algoritma *Knuth Morris Pratt* pada permainan Hanacaraka ditunjukkan pada gambar 3.3.



Gambar 3.3 Flowchart algoritma Knuth Morris Pratt

3.2.2.3 Perancangan Sistem

Perancangan sistem yang akan dibahas meliputi perancangan aplikasi permainan dan perancangan antarmuka.

1. Perancangan Aplikasi Permainan

Perancangan aplikasi permainan merupakan perancangan interaksi yang dapat dilakukan pemain terhadap aplikasi permainan yang dibangun. Aplikasi akan dimulai dengan menampilkan *scene* Menu yang tersedia tombol pilihan Mulai, Cara Main, dan Keluar. Setiap tombol akan memberikan tampilan pada *scene-scene* berikutnya. Aplikasi akan berhenti jika pengguna memilih tombol Keluar. Berikut ini adalah proses yang terjadi pada aplikasi *game* Hanacaraka yang dapat dilihat pada Gambar 3.4.

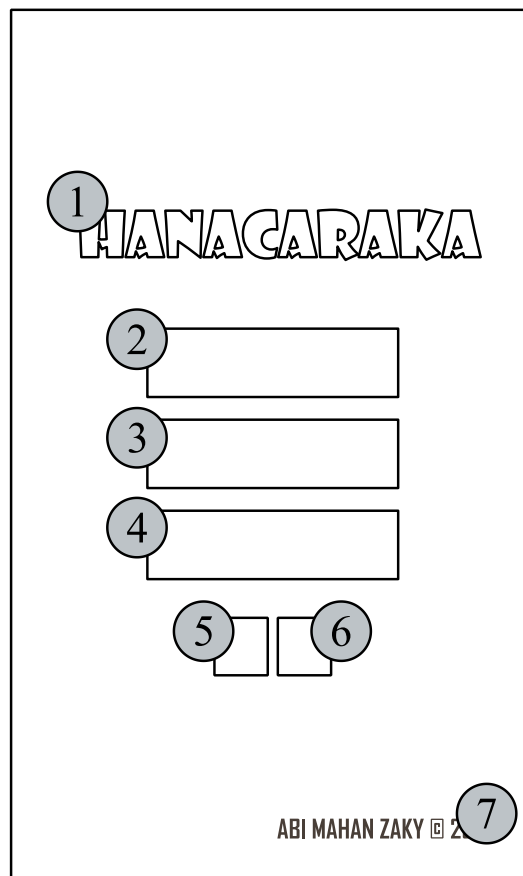
Gambar 3.4 *Flowchart* Aplikasi Permainan Hanacaraka

2. Perancangan Antarmuka

Perancangan antarmuka aplikasi permainan Hanacaraka ini terdiri dari beberapa *scene*, yaitu *scene* Menu, *scene* Pilih Level, *scene* Pilih Soal, *scene* Arena Permainan, dan *scene* Cara Main. Berikut ini adalah rancangan antarmuka *scene-scene* tersebut:

a. *Scene* Menu

Scene Menu merupakan *scene* yang pertama kali ditampilkan pada saat aplikasi dijalankan. *Scene* ini berisikan menu yang dapat dipilih oleh pengguna yaitu Mulai Main, Cara Main, dan Keluar. Rancangan antarmuka *scene* ini dapat dilihat pada gambar 3.5.



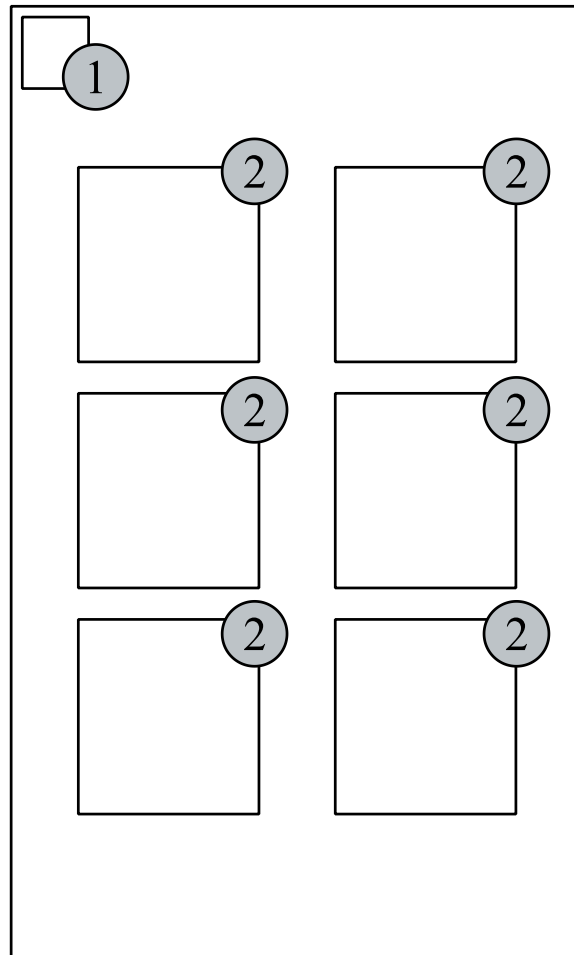
Gambar 3.5 Rancangan *Scene* Menu

Keterangan:

- 1) Judul Hanacaraka merupakan judul aplikasi yang dirancang dengan menggunakan komponen *graphic*.
- 2) Tombol Mulai Main merupakan tombol yang berfungsi untuk menampilkan *scene* Arena Permainan yang dirancang dengan menggunakan komponen *button*.
- 3) Tombol Cara Main merupakan tombol yang berfungsi untuk menampilkan *scene* Cara Main yang dirancang dengan menggunakan komponen *button*.
- 4) Tombol Keluar merupakan tombol yang berfungsi untuk keluar dari *game* Hanacaraka yang dirancang dengan menggunakan komponen *button*.
- 5) Tombol Twitter merupakan tombol yang berfungsi untuk membuka halaman web www.twitter.com/abimahan yang dirancang dengan menggunakan komponen *button*.
- 6) Tombol Facebook merupakan tombol yang berfungsi untuk membuka halaman web www.facebook.com/abimahan yang dirancang dengan menggunakan komponen *button*.
- 7) Tulisan Abi Mahan Zaky © 2104 merupakan nama pembuat aplikasi dan tahun dibuatnya aplikasi yang dirancang dengan menggunakan komponen *text*.

b. *Scene* Pilih Level

Scene Pilih Level merupakan *scene* yang berisikan pilihan level yang dapat dipilih untuk dimainkan oleh *user*. Rancangan antarmuka *scene* ini dapat dilihat pada gambar 3.6.



Gambar 3.6 Rancangan *Scene* Pilih Level

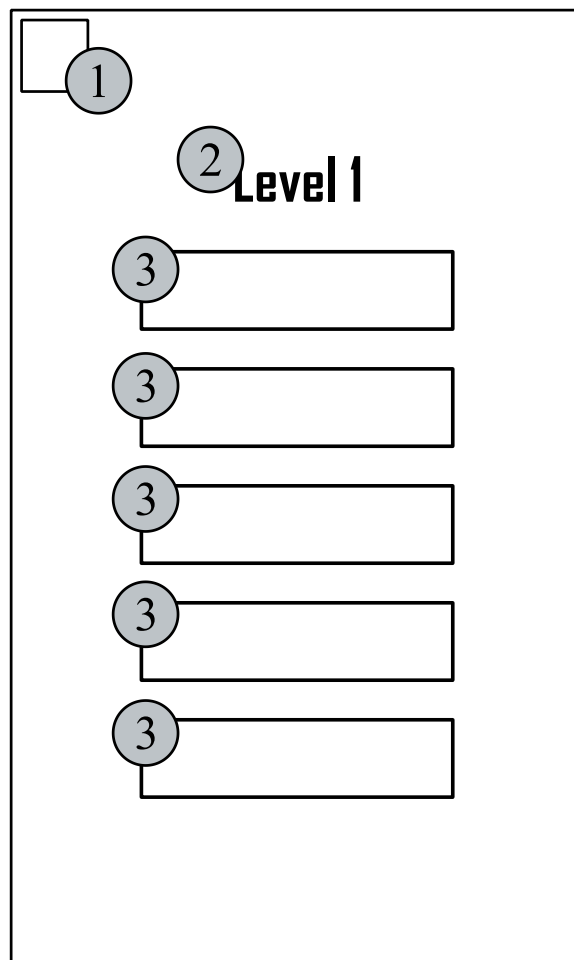
Keterangan:

- 1) Tombol Kembali merupakan tombol yang berfungsi untuk menampilkan *scene* sebelumnya yaitu *scene* Menu yang dirancang dengan menggunakan komponen *button*.

2) Tombol Level merupakan tombol yang berfungsi untuk menampilkan *scene* Pilih Soal yang dirancang dengan menggunakan komponen *button*.

c. *Scene* Pilih Soal

Scene Pilih Soal merupakan *scene* yang berisikan pilihan soal yang dapat dipilih untuk dimainkan oleh *user*. Rancangan antarmuka *scene* ini dapat dilihat pada gambar 3.7.



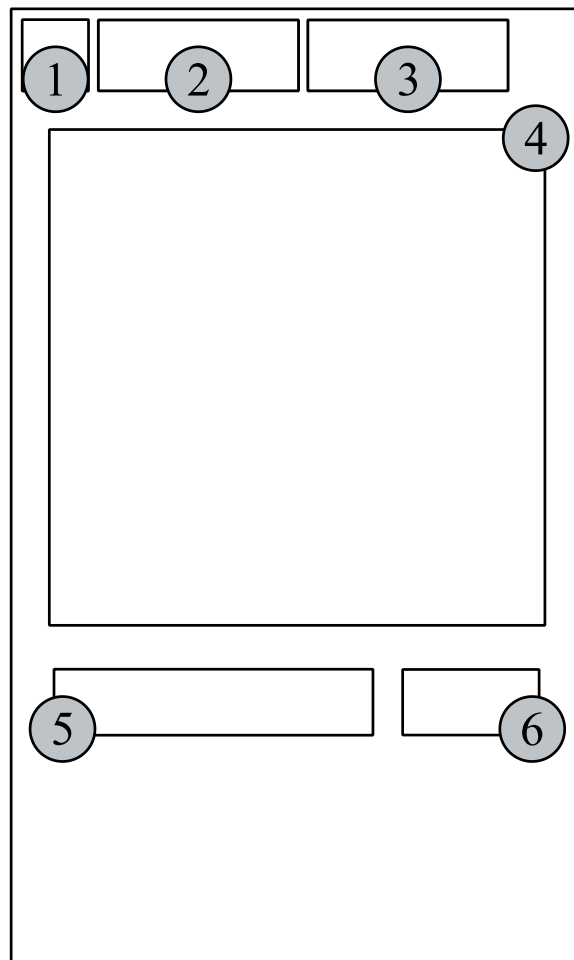
Gambar 3.7 Rancangan *Scene* Pilih Soal

Keterangan:

- 1) Tombol Kembali merupakan tombol yang berfungsi untuk menampilkan *scene* sebelumnya yaitu *scene* Pilih Level yang dirancang dengan menggunakan komponen *button*.
- 2) Tulisan Level merupakan tulisan yang menunjukkan level yang akan dimainkan yang dirancang dengan menggunakan komponen *button*.
- 3) Tombol Soal merupakan tombol yang berfungsi untuk menampilkan *scene* Arena Permainan yang dirancang dengan menggunakan komponen *button*.

d. *Scene* Arena Permainan

Scene Arena Permainan merupakan *scene* yang digunakan untuk bermain *game* Hanacaraka. Rancangan antarmuka *scene* ini dapat dilihat pada gambar 3.8.



Gambar 3.8 Rancangan *Scene* Arena Permainan

Keterangan:

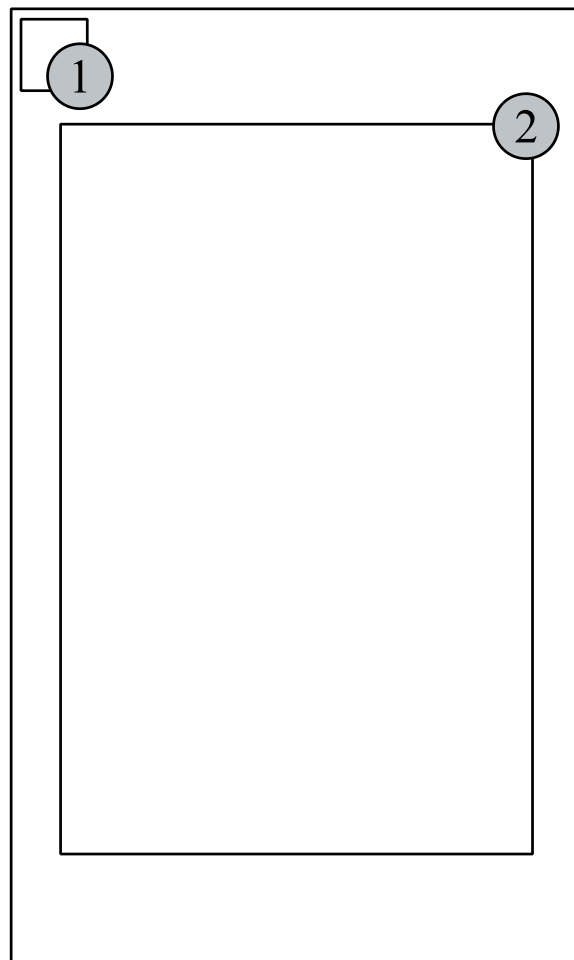
- 1) Tombol Kembali merupakan tombol yang berfungsi untuk menampilkan *scene* sebelumnya yaitu *scene* Pilih Soal yang dirancang dengan menggunakan komponen *button*.
- 2) Kotak Nyawa merupakan kotak yang berfungsi untuk menampilkan jumlah nyawa yang dimiliki pemain yang dirancang dengan menggunakan komponen *graphic* dan *text*.
- 3) Kotak Bantuan merupakan kotak yang berfungsi untuk menampilkan jumlah bantuan yang dimiliki oleh pemain dan juga

apabila di klik akan memunculkan bantuan yang berfungsi untuk membantu pemain dalam menyelesaikan soal. Kotak Bantuan ini dirancang dengan menggunakan Komponen *button* dan *text*.

- 4) Papan Soal merupakan kotak yang berfungsi untuk menampilkan soal dari permainan yang dirancang dengan menggunakan komponen *graphic*.
- 5) Kotak Jawaban merupakan tempat yang digunakan untuk mengetikkan jawaban dari soal yang dirancang dengan menggunakan komponen *input text*.
- 6) Tombol Cek merupakan tombol yang digunakan untuk memeriksa jawaban yang telah dimasukkan pemain pada Kotak Jawaban yang dirancang dengan menggunakan komponen *button*.

e. *Scene* Cara Main

Scene Cara Main merupakan *scene* yang berisikan peraturan, cara main, sekaligus materi tentang aksara Jawa, sehingga dapat memberikan informasi kepada pengguna tentang *game* Hancaraka dan tentang materi aksara Jawa. Rancangan antarmuka *scene* ini dapat dilihat pada gambar 3.9.



Gambar 3.9 Rancangan *Scene* Cara Main

Keterangan:

- 1) Tombol Home merupakan tombol yang digunakan untuk menampilkan *scene* Menu yang dirancang dengan menggunakan komponen *button*.
- 2) Kotak Aturan merupakan kotak yang digunakan untuk menampilkan aturan permainan yang dirancang dengan menggunakan komponen *text* dan *graphic*.

3.2.2.4 Uji Coba Sistem

Pada penelitian ini peneliti menggunakan pengujian *blackbox* sebagai uji coba produk. Pengujian menggunakan *blackbox testing* merupakan teknik pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, *blackbox testing* atau pengujian *blackbox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *blackbox* bukan merupakan alternatif dari teknik pengujian *whitebox*, tetapi merupakan pendekatan yang mampu mengungkapkan kelas kesalahan daripada metode pengujian *whitebox* (Pressman, 2002: 551).

Pengujian *blackbox* berusaha menemukan kesalahan dalam kategori sebagai berikut: (1) fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan *interface*, (3) kesalahan dalam struktur data atau akses database eksternal, (4) kesalahan kinerja, serta (5) inisialisasi dan kesalahan terminasi (Pressman, 2002: 551-552).

Dalam pembuatan lembar pengujian *blackbox*, peneliti menuliskan semua fungsi dan cara mengeksekusi kemudian meminta pengguna untuk mencoba aplikasi *game* Hanacaraka. Apabila ada tambahan yang ingin disarankan pengguna bisa mengisikan saran pada kolom keterangan yang sudah tersedia pada setiap fungsi. Berikut ini adalah skenario format pengujian *blackbox* yang akan diujikan kepada pengguna perangkat lunak dapat dilihat pada tabel 3.3.

Tabel 3.3. Skenario Pengujian *Blackbox*

No	Fungsi	Skenario	Hasil	Keterangan
(a)	(b)	(c)	(d)	(e)

Keterangan:

- (a) Nomor urut fungsi.
- (b) Nama tombol menu atau fitur aplikasi *game* Hanacaraka.
- (c) Skenario atau tata cara dalam menjalankan tombol atau fitur dalam perangkat lunak.
- (d) Hasil pengujian yang menjelaskan apakah fungsi dapat dapat dijalankan/ digunakan atau tidak.
- (e) Saran atau komentar oleh penguji perangkat lunak.

3.3 Teknik Pengumpulan Data

Pada penelitian ini peneliti menggunakan metode angket (*checklist*) dalam teknik pengumpulan datanya. Angket atau Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya (Sugiyono 2009:142). Angket digunakan dalam penelitian ini, tujuannya adalah untuk mengukur indikator fungsionalitas perangkat lunak.

Angket yang digunakan, mengacu pada format 5 *point* dari skala *likert*. Skala *likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Sugiyono 2009:93). Melalui

skala *likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai acuan untuk menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan.

3.4 Teknik Analisis Data

Menurut Wina Sanjaya (2009: 106), analisis data adalah suatu proses mengolah dan menginterpretasikan data dengan tujuan untuk mendudukkan berbagai informasi sesuai dengan fungsinya sehingga memiliki makna. Analisis data dapat dilakukan melalui tiga tahap yaitu mereduksi data, mendeskripsikan data dan membuat kesimpulan.

1. Mereduksi data merupakan kegiatan menyeleksi data sesuai dengan fokus permasalahan. Pada tahap ini peneliti mengumpulkan semua instrumen yang digunakan untuk mengumpulkan data untuk dikelompokkan sesuai masalah. Hal ini juga memungkinkan peneliti untuk membuang data yang tidak diperlukan.
2. Mendeskripsikan data dilakukan agar data yang telah diorganisir menjadi bermakna. Bentuk deskripsi tersebut dapat berupa naratif, grafik atau dalam bentuk tabel.
3. Tahap terakhir adalah membuat kesimpulan dari data yang telah dideskripsikan. Tahap menganalisis dan menginterpretasikan data merupakan tahap yang paling penting karena hal ini untuk memberikan makna dari data yang telah dikumpulkan. Hasil analisis dan interpretasi data merupakan jawaban dari rumusan masalah yang telah ditentukan sebelumnya.

Analisis data dalam penelitian ini menggunakan teknik analisis deskriptif kualitatif serta kuantitatif. Penghitungan data kuantitatif adalah dengan menghitung rata-rata skor yang diperoleh dari lembar angket yang telah disusun sebelumnya. Dengan rata-rata yang diperoleh dapat diketahui persentase fungsionalitas perangkat lunak. Adapun cara menghitung hasil (skor) yang diperoleh dengan rumus mean atau rerata nilai menurut Suharsimi Arikunto (2010: 284-285) yaitu sebagai berikut:

$$\bar{x} = \frac{\sum X}{N}$$

Keterangan :

$\bar{x}$ = Mean (rata-rata)

$\sum X$ = Jumlah keseluruhan nilai

N = Jumlah data dari nilai yang akan dirata-rata

Suharsimi Arikunto (2010: 269) menjelaskan analisis data deskriptif kualitatif adalah analisis data yang menggunakan teknik deskriptif kualitatif memanfaatkan persentase merupakan langkah awal saja dari keseluruhan proses analisis. Persentase yang dinyatakan dalam bilangan sudah jelas merupakan ukuran yang bersifat kuantitatif, bukan kualitatif. Jadi pernyataan persentase bukan hasil analisis kualitatif. Analisis kualitatif tentu harus dinyatakan dalam sebuah predikat yang menunjuk pada pernyataan keadaan, ukuran kualitas.

Berdasarkan pendapat di atas agar diperoleh hasil analisis kualitatif maka dari perhitungan persentase kemudian dimasukkan ke dalam lima kategori predikat. Menurut Suharsimi Arikunto (2010: 269) lima kategori predikat tersebut yaitu seperti pada tabel 3.4.

Tabel 3.4. Kategori Predikat Fungsionalitas Perangkat Lunak

No	Interval	Kategori
1.	81-100%	Sangat Layak
2.	61-80%	Layak
3.	41-60%	Cukup
4.	21-40%	Kurang Layak
5.	0-20%	Tidak Layak

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dapat diambil kesimpulan telah berhasil mengimplementasikan algoritma *Knuth Morris Pratt* pada perancangan *game* Hanacaraka. Hasil pengujian pada perangkat lunak diperoleh presentase 97,14% pada pengujian fungsionalitas perangkat lunak. Dari skor presentase yang didapat maka kualitas perangkat lunak telah mempunyai predikat Sangat Layak. Kualitas pencocokan *string* dalam *game* berhasil ditingkatkan dengan adanya pengecekan jawaban yang hampir benar dimana tidak bisa dilakukan jika tidak menggunakan algoritma *Knuth Morris Pratt*.

5.2 Saran-saran

Penelitian perancangan *game* Hanacaraka memiliki beberapa kekurangan sehingga dapat disarankan untuk penelitian lanjutan sebagai berikut:

1. Perlu adanya pengembangan lebih lanjut untuk menambah animasi yang ada karena animasi yang ada pada *game* masih terlalu kaku.
2. Perlu adanya penambahan variasi dan jumlah soal karena variasi dan jumlah soal yang ada masih terlalu sedikit.
3. Perlu adanya penambahan fungsi untuk menonaktifkan suara yang ada pada *game* karena suara yang ada pada *game* belum bisa dimatikan kecuali volume suara dari perangkat tersebut yang dinonaktifkan.

4. Perlu adanya manajemen *asset* yang baik pada *game* karena ukuran *game* masih relatif terlalu besar untuk *game* sejenis.
5. Perlu adanya penambahan fitur pada *game* seperti *timer* agar permainan lebih menantang.

DAFTAR PUSTAKA

- Crawford, C.2003. *Chris Crawford on Game Design*. Upper Saddle River: New Rider Publishing.
- Dastbaz, M. 2003. *Designing Interactive Multimedia*. New York: McGrawHill.
- Fowler, M. 1996. *UML Distilled Second Edition : A Brief to The Standard Object Modelling Language.Second Edition*. New Jersey:Pearson Education.
- Gallagher, D. (2013), *7 health benefits of playing video games*. Available at <http://www.theweek.com/> [accessed 1/10/2014].
- H., Nazruddin Safaat. 2012. *ANDROID: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Informatika Bandung.
- Kramer, W. (2006). *What Makes a Game Good?*. Available at www.thegamesjournal.com [accessed 1/12/2014].
- Meier, Reto. 2009. *Professional Android™ Application Development* . Wiley Publishing, Inc.
- Meigs, T. 2003. *Ultimate Game Design : Building Game World*. California: McGraw-Hill.
- Munir, Rinaldi.Diktat Kuliah IF2251 *Strategi Algoritmik*. Institut Teknologi Bandung. 2007.
- Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak :Pendekatan Praktisi (Buku 1)*. Yogyakarta : Andi
- Putra, A. R. P. P., Djuniadi. 2013. MEMPERCANTIK TAMPILAN GAME 2D MENJADI 3D DE KRONIK VAN DIPONEGORO MENGGUNAKAN TEKNIK MODE 7. *Jurnal Teknik Elektro Unnes*.5(2):102-106.
- Rollings, A. & A. Ernest. 2003. *On Game Design*. Upper Saddle River: New Riders Publishing.
- Schell, J. 2008. *The Art of Game Design a Book of Lenses. 1st Edition*. Boca Raton, Florida: CRC Press.
- Sugiyono.2009.*Metode Penelitian Pendidikan*.Bandung:Alfabeta.

- Sugiyono, Prof. Dr. .2010. *Statistika untuk Penelitian*. Bandung : Alfabeta.
- Suharsimi Arikunto 2010. *Prosedur Penelitian Suatu Pendekatan Praktik*. Yogyakarta : Rineka Cipta.
- Whitten, J. L.& L. D. Bentley. 2007. *Systems Analysis and Design Methods*. New York: McGrawHill.
- Wina Sanjaya. (2009). *Strategi Pembelajaran Berorientasi Standar Proses Pendidikan*. Jakarta: Kencana Prenada Media Group

Lampiran 1

SURAT KELENGKAPAN



• **KEPUTUSAN
DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG**

Nomor: 688/FT-UNNES/2014

Tentang

**PENETAPAN DOSEN PEMBIMBING SKRIPSI/TUGAS AKHIR SEMESTER
GASAL/GENAP
TAHUN AKADEMIK 2014/2015**

- Menimbang : Bahwa untuk memperlancar mahasiswa Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Fakultas Teknik membuat Skripsi/Tugas Akhir, maka perlu menetapkan Dosen-dosen Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Fakultas Teknik UNNES untuk menjadi pembimbing.
- Mengingat : 1. Undang-undang No.20 Tahun 2003 tentang Sistem Pendidikan Nasional (Tambahan Lembaran Negara RI No.4301, penjelasan atas Lembaran Negara RI Tahun 2003, Nomor 78)
2. Peraturan Rektor No. 21 Tahun 2011 tentang Sistem Informasi Skripsi UNNES
3. SK. Rektor UNNES No. 164/O/2004 tentang Pedoman penyusunan Skripsi/Tugas Akhir Mahasiswa Strata Satu (S1) UNNES;
4. SK Rektor UNNES No.162/O/2004 tentang penyelenggaraan Pendidikan UNNES;
- Menimbang : Usulan Ketua Jurusan/Prodi Teknik Elektro/Pend. Teknik Informatika dan Komputer Tanggal 2 September 2014

MEMUTUSKAN

Menetapkan :

PERTAMA :

Menunjuk dan menugaskan kepada:

Nama : Riana Defi Mahadji Putri, ST, MT

NIP : 197609182005012001

Pangkat/Golongan : III/B

Jabatan Akademik : Asisten Ahli

Sebagai Pembimbing

Untuk membimbing mahasiswa penyusun skripsi/Tugas Akhir :

Nama : ABI MAHAN ZAKY

NIM : 5302410208

Jurusan/Prodi : Teknik Elektro/Pend. Teknik Informatika dan Komputer

Topik : Implementasi Algoritma Knuth Morris Pratt Pada Perancangan Game Hanacaraka

KEDUA :

Keputusan ini mulai berlaku sejak tanggal ditetapkan.

Tembusan

1. Pembantu Dekan Bidang Akademik
2. Ketua Jurusan
3. Petinggal

5302410208
73-AKD-24/Rev. 00

DITETAPKAN DI : SEMARANG
PADA TANGGAL : 2 September 2014
DEKAN

UNNES
Drs. Muhammad Harjanu, M.Pd.
NIP. 195602151991021001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI SEMARANG
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
Gedung E6 Lt 2, Kampus Sekaran, Gunungpati, Semarang 50229
Telepon: 8508104
Laman: www.te.unnes.ac.id, surel:

Nomor : 266/TE/IX/2014
Lamp. :
Hal : Usulan Pembimbing

Yth. Dekan Fakultas Teknik
Universitas Negeri Semarang

Merujuk Keputusan Rektor Unnes Nomor 164/O/2004 tentang Pedoman Penyusunan Skripsi Mahasiswa Program S1 pasal 7 mengenai penentuan pembimbing, dengan ini saya usulkan

Nama : Riana Defi Mahadji Putri, ST, MT
NIP : 197609182005012001
Pangkat/Golongan : III/B
Jabatan Akademik : Asisten Ahli
Sebagai Dosen Pembimbing

Dalam penyusunan Skripsi/Tugas Akhir untuk mahasiswa

Nama : ABI MAHAN ZAKY
NIM : 5302410208
Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
Topik : Implementasi Algoritma Knuth Morris Pratt Pada Perancangan Game Hanacaraka
Untuk itu, mohon diterbitkan surat penetapannya.



Semarang, 2 September 2014

Ketua Jurusan

Drs. Buryono, M.T.

NIP. 195503161985031001





Formulir Usulan Topik Skripsi
FM-1-AKD-24/rev.00
UNIVERSITAS NEGERI SEMARANG

Usulan topik skripsi ini diajukan oleh:

Nama : ABI MAHAN ZAKY
NIM : 5302410208
Jurusan : Teknik Elektro
Program Studi : Pendidikan Teknik Informatika dan Komputer, S1
Topik : Implementasi Algoritma Knuth Morris Pratt Pada Perancangan Game Hanacaraka



Menyetujui
Ketua Jurusan

Drs. Suryono, M.T.
NIP. 195503161985031001

Semarang, 1 September 2014
Yang mengajukan,


ABI MAHAN ZAKY
NIM. 5302410208





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI SEMARANG

FAKULTAS TEKNIK

Gedung E6 Lt 2, Kampus Sekaran, Gunungpati, Semarang 50229
Telepon: 8508104
Laman: www.te.unnes.ac.id, surel:

No. : 1447/UM.37.1-S/DT/2015
Lamp. :
Hal : Surat Tugas Panitia Ujian Sarjana

Dengan ini kami tetapkan bahwa ujian Sarjana Fakultas Teknik UNNES untuk jurusan Teknik Elektro adalah sebagai berikut:

I. Susunan Panitia Ujian:

- | | |
|---------------------|---|
| a. Ketua | : Drs. Suryono, M.T. |
| b. Sekretaris | : FEDDY SETIO PRIBADI, S.Pd., MT. |
| c. Pembimbing Utama | : Riana Defi Mahadji Putri, ST, MT |
| d. Penguji | : 1. FEDDY SETIO PRIBADI, S.Pd., MT.
2. ARYO BASKORO UTOMO, S.T., M.T. |

II. Calon yang diuji:

- | | |
|---------------------------|---|
| Nama | : ABI MAHAN ZAKY |
| NIM/Jurusan/Program Studi | : 5302410208/Teknik Elektro
(Pendidikan Teknik Informatika dan Komputer, S1) |
| Judul Skripsi | : Implementasi Algoritma Knuth Morris Pratt Pada Perancangan Game Mancaraka |

III. Waktu dan Tempat Ujian:

- | | |
|--------------|----------------------------|
| Hari/Tanggal | : Senin / 23 Februari 2015 |
| Jam | : 11:00:00 |
| Tempat | : E6 244 |
| Pakaian | : |

Tembusan
1. Ketua Jurusan Teknik Elektro
2. Calon yang diuji



23 - 2 - 2015

Mubammad Harlanu, M.Pd.
NIP.196602151991021001

5302410208

Lampiran 2

FORM ANGKET (CHECKLIST)

Nama :

Asal Instansi :

Status :

Petunjuk :

1. Isilah nama dan asal instansi Anda pada kolom yang disediakan
2. Cobalah Fungsi-fungsi yang terdapat pada *game* Hanacaraka sesuai dengan skenario yang ada
3. Apabila skenario yang dilakukan berhasil maka isikan “v” pada kolom Ya dan apabila hasil tidak sesuai maka isikan “v” pada kolom Tidak

TEST CASE FUNCTION

No	Fungsi	Pertanyaan	Ya	Tidak
1.	Koreksi Jawaban	Apakah bila jawaban yang dimasukkan benar fungsi koreksi jawaban benar sudah berjalan dengan benar?		
2.	Koreksi Jawaban	Apakah bila jawaban yang dimasukkan salah fungsi koreksi jawaban salah sudah berjalan dengan benar?		
3.	Koreksi Jawaban	Apakah bila jawaban yang dimasukkan hampir benar fungsi koreksi jawaban hampir sudah berjalan dengan benar?		
4.	Fungsi Bantuan	Apakah bila tombol bantuan ditekan fungsi bantuan akan aktif?		
5.	Fungsi <i>Lock</i> Level	Apakah fungsi <i>lock</i> level sudah berjalan dengan benar?		
6.	Fungsi <i>Lock</i> Soal	Apakah fungsi <i>lock</i> soal sudah berjalan dengan benar?		
7.	Fungsi Medali	Apakah fungsi medali sudah berjalan dengan benar?		

Semarang,

2014

Lampiran 3

CONTOH TABEL PENGKOREKSIAN KATA

Kata yang diuji	Pengkoreksi Kata
Hanacaraka	anacaraka hnacaraka haacaraka hancaraka hanaaraka hanacraka hanacaaka hanacarka hanacaraa hanacarak ahnacaraka hnaacaraka haancaraka hancaaraka hanaacraka hanacraaka hanacaarka hanacarkaa hanacaraak aanacaraka banacaraka canacaraka danacaraka eanacaraka fanacaraka ganacaraka hanacaraka ianacaraka janacaraka kanacaraka lanacaraka manacaraka nanacaraka oanacaraka panacaraka qanacaraka ranacaraka sanacaraka tanacaraka uanacaraka vanacaraka wanacaraka xanacaraka yanacaraka zanacaraka hbnacaraka hcnacaraka hdnacaraka henacaraka hfnacaraka hgnacaraka hhnacaraka hinacaraka hjnacaraka hknacaraka hlnacaraka hmnacaraka hnnacaraka honacaraka hpnacaraka hqnacaraka hrnacaraka hsnacaraka htnacaraka hunacaraka hvnacaraka hwnacaraka hxnacaraka hynacaraka hznacaraka haaacaraka habacaraka hacacaraka hadacaraka haeacaraka hafacaraka hagacaraka hahacaraka haiacaraka hajacaraka hakacaraka halacaraka hamacaraka haoacaraka hapacaraka haqacaraka haracaraka hasacaraka hatacaraka hauacaraka havacaraka hawacaraka haxacaraka hayacaraka hazacaraka hanbcaraka hancaraka handcaraka hanecaraka hanfcaraka hangcaraka hanhcaraka hanicaraka hanjcaraka hankcaraka hanlcaraka hanmcaraka hanncaraka hanocaraka hanpcaraka hanqcaraka hanrcaraka hanscaraka hantcaraka hanucaraka hanvcaraka hanwcaraka hanxcaraka hanycaraka hanzcaraka hanaaaraka hanabaraka hanadaraka hanaearaka hanafaraka hanagaraka hanaharaka hanaiaraka hanajaraka hanakaraka hanalaraka hanamaraka hananaraka hanaoaraka hanaparaka hanaqaraka hanararaka hanasaraka hanataraka hanauaraka hanavaraka hanawaraka hanaxaraka hanayaraka hanazaraka hanacbraka hanaccraka hanacdraka hanaceraka hanacfraka hanacgraka hanachraka hanaciraka hanacjraka hanackraka hanaclraka hanacmraka hanacnraka hanacoraka hanacpraka hanacqraka hanacrraka hanacsraka hanactraka hanacuraka hanacvraka hanacwraka hanacxraka hanacyraka hanaczraka hanacaaaka hanacabaka hanacacaka hanacadaka hanacaeaka hanacafaka hanacagaka hanacahaka hanacaiaka hanacajaka hanacakaka hanacalaka hanacamaka hanacanaka hanacaoaka hanacapaka hanacaqaka hanacasaka hanacataka hanacauaka hanacavaka hanacawaka hanacaxaka hanacayaka hanacazaka hanacarbka hanacarcka hanacardka hanacareka hanacarfka hanacargka hanacarhka hanacarika hanacarjka hanacarkka hanacarlka hanacarmka hanacarnka hanacaroka hanacarpka hanacargka hanacarrka hanacarska hanacartka hanacaruka hanacervka hanacarwka hanacarxka hanacaryka hanacarzka hanacaraaa hanacaraba hanacaraca hanacarada hanacaraea hanacarafa hanacaraga hanacaraha hanacaraia hanacaraja hanacarala

	hanacarama hanacarana hanacaraoa hanacarapa hanacaraqa hanacarara hanacarasa hanacarata hanacaraua hanacarava hanacarawa hanacaraxa hanacaraya hanacaraza hanacarakb hanacarakc hanacarakd hanacarake hanacarakf hanacarakg hanacarakh hanacaraki hanacarakj hanacarakk hanacarakl hanacarakm hanacarakn hanacarako hanacarakp hanacarakq hanacarakr hanacaraks hanacarakt hanacaraku hanacarakv hanacarakw hanacarakx hanacaraky hanacarakz ahanacaraka bhanacaraka chanacaraka dhanacaraka ehanacaraka fhanacaraka ghanacaraka hhanacaraka ihanacaraka jhanacaraka khanacaraka lhanacaraka mhanacaraka nhanacaraka ohanacaraka phanacaraka qhanacaraka rhanacaraka shanacaraka thanacaraka uhanacaraka vhanacaraka whanacaraka xhanacaraka yhanacaraka zhanacaraka haanacaraka hbanacaraka hcanacaraka hdanacaraka heanacaraka hfanacaraka hganacaraka hianacaraka hjanacaraka hkanacaraka hlanacaraka hmanacaraka hnanacaraka hoanacaraka hpanacaraka hqanacaraka hranacaraka hsanacaraka htanacaraka huanacaraka hvanacaraka hwanacaraka hxanacaraka hyanacaraka hzanacaraka habnacaraka hacnacaraka hadnacaraka haenacaraka hafnacaraka hagnacaraka hahnacaraka hainacaraka hajnacaraka haknacaraka halnacaraka hamnacaraka hannacaraka haonacaraka hapnacaraka haqnacaraka harnacaraka hasnacaraka hatnacaraka haunacaraka havnacaraka hawnacaraka haxnacaraka haynacaraka haznacaraka hanaacaraka hanbacaraka hancacaraka handacaraka haneacaraka hanfacaraka hangacaraka hanhacaraka haniacaraka hanjacaraka hankacaraka hanlacaraka hanmacaraka hanoacaraka hanpacaraka hanqacaraka hanracaraka hansacaraka hantacaraka hanuacaraka hanvacaraka hanwacaraka hanxacaraka hanyacaraka hanzacaraka hanabcaraka hanaccaraka hanadcaraka hanaecaraka hanafcaraka hanagcaraka hanahcaraka hanaicaraka hanajcaraka hanakcaraka hanalcaraka hanamcaraka hanancaraka hanaocaraka hanapcaraka hanaqcaraka hanarcaraka hanascaraka hanatcaraka hanaucaraka hanavcaraka hanawcaraka hanaxcaraka hanaycaraka hanazcaraka hanacaaraka hanacbaraka hanacdaraka hanacearaka hanacfaraka hanacgaraka hanacharaka hanaciaraka hanacjaraka hanackaraka hanaclaraka hanacmaraka hanacnaraka hanacoaraka hanacparaka hanacqaraka hanacraraka hanacsaraka hanactaraka hanacuaraka hanacvaraka hanacwaraka hanacxaraka hanacyaraka hanaczaraka hanacabraka hanacacraka hanacadraka hanacaeraka hanacafraka hanacagraka hanacahraka hanacairaka hanacajraka hanacakra hanacalraka hanacamraka hanacanraka hanacaoraka hanacapraka hanacaqaraka hanacarraka hanacasraka hanacatraka hanacauraka hanacavraka hanacawraka hanacaxraka hanacayraka hanacazraka hanacaraaka hanacarbaka hanacarcaka hanacardaka hanacareaka hanacarfaka hanacargaka hanacarthaka hanacariaka hanacarjaka hanacarkaka hanacarlaka hanacarmaka hanacarnaka hanacaroaka hanacarpaka hanacarqaka hanacarsaka hanacartaka hanacaruaka hanacarvaka hanacarwaka hanacarxaka hanacaryaka hanacarzaka hanacarabka hanacaracka hanacaradka hanacaraeka hanacarafka hanacaragka hanacarahka hanacaraika hanacarajka hanacarakka
--	--

	hanacaralka hanacaramka hanacaranka hanacaraoka hanacarapka hanacaraqka hanacararka hanacaraska hanacaratka hanacarauka hanacaravka hanacarawka hanacaraxka hanacarayka hanacarazka hanacarakaa hanacarakba hanacarakca hanacarakda hanacarakea hanacarakfa hanacarakga hanacarakha hanacarakia hanacarakja hanacarakla hanacarakma hanacarakna hanacarakoa hanacarakpa hanacarakqa hanacarakra hanacaraksa hanacarakta hanacarakua hanacarakva hanacarakwa hanacarakxa hanacarakya hanacarakza
--	--