

Spelling chekcer

Proyek 2



Alvian Daniel Sinaga 1184077

Gany Berdu Sura 1184008

In Partial Fulfilment of The Requirements for The Degree of Applied

Bachelor of Informatics Engineering

Program Studi D4 Teknik Informatika

Applied Bachelor Program of Informatics Engineering

Politeknik Pos Indonesia

Bandung 2018

‘Jika Kamu tidak dapat menahan lelahnya belajar, Maka
kamu harus sanggup menahan perihnya Kebodohan.’

Imam Syafi’i

LEMBAR PENGESAHAN

LEMBAR PERNYATAAN

Abstract

Abstract

Acknowledgements

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga Buku Pedoman Tingkat Akhir ini dapat diselesaikan.

DAFTAR ISI

COVER	i
LEMBAR PENGESAHAN LAPORAN	ii
LEMBAR PERNYATAAN	iii
HALAMAN PERNYATAAN	iv
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x

DAFTAR TABEL	xi
---------------------	-----------

DAFTAR SIMBOL	xi
----------------------	-----------

DAFTAR SINGKATAN	xiii
-------------------------	-------------

I Pendahuluan	1
----------------------	----------

1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	2
1.3 Rumusan Masalah	2
1.4 Tujuan	2
1.5 Manfaat Penelitian	2
1.6 Sistematika Penulisan	2

II Landasan Teori	4
--------------------------	----------

2.1 Algoritma	4
2.2 Sistem	4
2.3 Analisis	5
2.4 Aplikasi	5
2.5 bahasa pemograman	6
2.5.1 PHP	6
2.5.2 My SQL	7
2.6 database	7
2.7 Google API	7
2.8 BPMN	7

2.9	UML	8
2.9.1	Use Case Diagram	8
2.9.2	Activity Diagram.....	10
2.9.3	Sequence Diagram.....	10
2.9.4	Class Diagram.....	11
2.10	BlackBox Testing.....	11
III	Analisis Dan Perancangan Sistem	13
3.1	Analisis Sistem.....	13
3.1.1	Analisis Sistem Yang Sedang Berjalan (Current System)	13
3.1.1.1	Analisis BPMN yang sedang berjalan.....	13
3.1.1.2	Analisis BPMN Yang Sedang Berjalan Pada Proses Copy Kata	14
3.1.1.3	Analisis Yang Sedang Berjalan Pada Proses Hasil Cek Karya Ilmiah	15
3.1.1.4	Analisis dokumen yang digunakan	16
3.2	Analisis Sistem yang akan Dibangun	16
3.2.0.1	Analisis Sistem yang akan Dibangun Pada Sub Proses Log in	17
3.3	Activity Diagram.....	18
3.4	Sequences Diagram.....	19
3.5	Class Diagram.....	20
IV	Metodologi Penelitian	21

4.1	Lingkungan Implementasi	21
4.1.1	Lingkungan perangkat lunak	21
4.2	Tempat dan Waktu Penelitian	22
4.3	Jenis dan Sumber Data	23
4.4	Teknik Pengumpulan Data.....	23
4.5	Tahap Perencanaan	23
4.6	Tahap Pengumpulan Data	24
V	Conclusion	25
5.1	Conclusion of Problems	25
5.2	Conclusion of Method	25
5.3	Conclusion of Experiment	25
5.4	Conclusion of Result	25
VI	Kesimpulan Dan Saran	26
6.1	Kesimpulan	26
6.2	Saran.....	26
A	Form Penilaian Jurnal	27
B	FAQ	28
	DAFTAR PUSTAKA	30

DAFTAR GAMBAR

A.1	Form nilai bagian 1.....	27
A.2	form nilai bagian 2.	27

DAFTAR TABEL

DAFTAR SIMBOL

DAFTAR SINGKATAN

BAB I

Pendahuluan

1.1 Latar Belakang Masalah

Bahasa merupakan salah satu komponen penting pada tingkat pendidikan. Dalam bentuk tulisan, bahasa menjadi faktor penting dalam penyampaian pengetahuan dari generasi ke generasi. Bahasa juga menjadi acuan dalam penulisan dokumen, komunikasi dan pencarian informasi hingga tugas pada kalangan orang yang sedang mengenyam pendidikan. Dan hal ini Apabila dalam penulisan dokumen terdapat kesalahan maksud penulisan, pencarian atau komunikasi tersebut menjadi berbeda. Oleh karena itu, dibutuhkan sebuah program yang dapat mendeteksi kesalahan penulisan dan memberikan sugesti kata yang benar.

Jurusan Teknik Informatika Politeknik Pos Indonesia merupakan sebuah lembaga dibidang akademik yang telah menerapkan pembangunan sistem. Salah satu contoh sistem yang dibangun adalah Aplikasi Spelling checker. Namun, semakin canggih perkembangan teknologi harus berbanding lurus dengan aplikasi yang semakin maju pula.

Dalam studi kasus ini, tim pembuat aplikasi Spelling checker membutuhkan suatu analisis sistem dan fungsi yang dapat mempermudah penyampaian informasi kepada pengembang aplikasi agar dapat mengetahui kelebihan dan kekurangan dari sistem aplikasi yang telah dikembangkan. Aplikasi Pengecekan Plagiarisme tersebut terdiri dari beberapa proses bisnis yang dibuat menggunakan BPMN dan UML sebagai salah satu cara

menganalisis aplikasi Spelling checker yang akan dibangun.

BPMN dan UML merupakan cara menganalisis suatu sistem untuk membuat laporan mengenai sistem yang telah dibuat. Awalnya, BPMN menjelaskan sistem yang sedang berjalan dan yang akan dibangun pada aplikasi sesuai dengan standar internasional. Sedangkan UML menjelaskan secara spesifik mengenai aplikasi yang sedang berjalan dan yang akan dibangun. Hal ini sudah memenuhi standart internasional dalam melakukan analisis sistem sehingga membantu pembangunan aplikasi lebih teratur.

Oleh karena itu, pada penelitian ini penulis akan melakukan pembangunan aplikasi Spelling checker guna membantu mengurangi kesalahan dalam penulisan pada suatu

dokumen dan membantu memudahkan dalam hal pengecekan kesalahan penulisan tersebut.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan di atas maka identifikasi masalah tersebut adalah Membangun aplikasi Spelling checker .

1.3 Rumusan Masalah

bagaimana membuat aplikasi yang dapat mengurangi kesalahan dalam penulisan sebuah dokumen.

1.4 Tujuan

untuk mengurangi kesalahan dalam penulisan sebuah dokumen maka penulis membuat aplikasi Spelling.

1.5 Manfaat Penelitian

1. Bagi pengguna manfaat penelitian ini adalah mempermudah pengguna untuk melakukan pengecekan dokumen atau penulisan apapun dengan mudah dan efisien
2. Bagi peneliti manfaat penelitian ini adalah untuk mengembangkan sistem pengecekan Spell sehingga sistem ini dapat diimplementasikan dimana saja oleh pengguna.

1.6 Sistematika Penulisan

BAB I : Membahas tentang latar belakang, identifikasi masalah, rumusan masalah, tujuan penulisan, manfaat penulisan, kontribusi penelitian, serta sistematika penulisan

BAB II : Membahas tentang tinjauan pustaka yang mendukung dalam penulisan laporan

BAB III : Membahas tentang perencanaan aplikasi

BAB IV : Membahas tentang pengujian aplikasi dan analisis dari aplikasi yang sudah dibuat.

BAB V : Membuat kesimpulan dan saran untuk pengembangan lebih lanjut dari sistem yang telah dibuat

BAB II

Landasan Teori

2.1 Bahasa

Bahasa adalah alat komunikasi antara anggota masyarakat berupa simbol bunyi yang dihasilkan oleh alat ucap manusia. Mungkin ada yang keberatan dengan mengatakan bahwa bahasa bukan satu-satunya alat untuk mengadakan komunikasi. Mereka menunjukkan bahwa dua orang atau pihak yang mengadakan komunikasi dengan menggunakan cara-cara tertentu yang telah disepakati bersama. Lukisan-lukisan, asap api, bunyi gendang atau tong-tong dan sebagainya.

Tetapi mereka itu harus mengakui pula bahwa bila dibandingkan dengan bahasa, semua alat komunikasi tadi mengandung banyak segi yang lemah. Bahasa adalah sebuah sistem, artinya, bahasa itu dibentuk oleh sebuah komponen yang berpola secara tetap dan dapat dikaidahkan. Bagi orang yang mengerti sistem bahasa Indonesia akan mengakui bahwa susunan “Ibu meng...seekor...di...” adalah sebuah kalimat bahasa Indonesia yang benar sistemnya, meskipun ada sejumlah komponennya yang ditanggalkan. Tetapi susunan “Meng ibu se ikan goreng di dapur” bukanlah kalimat bahasa Indonesia yang benar karena tidak tersusun menurut sistem kalimat bahasa Indonesia. Sebagai sebuah sistem, bahasa selain bersifat sistematis juga bersifat sistemis. Dengan sistematis maksudnya, bahasa itu

tersusun menurut suatu pola tertentu, tidak tersusun secara acak atau sembarangan. Sistemis, artinya sistem bahasa itu bukan merupakan sebab sistem tunggal, melainkan terdiri atas sejumlah subsistem, yakni sub-sistem fonologi, subsistem morfologi, sub-sistem sintaksis, dan sub-sistem leksikon. Bahasa memberikan kemungkinan yang jauh lebih luas dan kompleks daripada yang dapat diperoleh dengan menggunakan media.

Bahasa merupakan bunyi yang dihasilkan oleh alat ucap manusia bukannya sembarang bunyi, melainkan bunyi itu sendiri haruslah merupakan simbol atau perlambang.

Pengertian Bahasa Bahasa merupakan aspek yang begitu penting dalam kehidupan bermasyarakat. Mengutip pengertian bahasa menurut pendapat Keraf yang menyatakan ada dua pengertian bahasa.

Pengertian pertama menyatakan bahwa bahasa sebagai alat komunikasi antara anggota masyarakat berupa simbol bunyi yang dihasilkan oleh alat ucap manusia.

Kedua, bahasa adalah sistem komunikasi yang menggunakan simbol-simbol vokal (bunyi ujaran) yang bersifat arbitrer (Suyanto, 2011: 15).

Tarigan memberikan dua definisi bahasa. Pertama, bahasa adalah suatu sistem yang sistematis, barangkali juga untuk sistem generatif. Kedua, bahasa adalah seperangkat lambang-lambang manasuka atau simbol-simbol arbitrer (Suyanto, 2011: 15).

Dalam Wikipedia, dijelaskan bahwa bahasa adalah alat atau perwujudan

budaya yang digunakan manusia untuk saling berkomunikasi atau berhubungan, baik lewat tulisan, maupun lisan, dengan tujuan menyampaikan maksud hati atau kemauan kepada lawan bicaranya atau orang lain.

Bahasa adalah keterampilan khusus yang kompleks, berkembang dalam diri anak-anak secara spontan, tanpa usaha sadar atau instruksi formal, dipakai tanpa memahami logika yang mendasarinya, secara kualitatif sama dalam diri setiap orang, dan berbeda dari kecakapan-kecakapan lain yang sifatnya lebih umum dalam memproses informasi atau berperilaku secara cerdas (Brown, 2008: 6).

Berdasarkan beberapa pengertian bahasa yang dikemukakan oleh para ahli di atas, penulis bisa mengambil kesimpulan bahwa bahasa adalah rangkaian sistem bunyi atau simbol yang dihasilkan oleh alat ucap manusia, yang memiliki makna dan secara konvensional digunakan oleh sekelompok manusia (penutur) untuk berkomunikasi (melahirkan pikiran dan perasaan) kepada orang lain.

Ada tiga pandangan tentang hakikat bahasa, seperti yang dikemukakan Jack C. Richards yaitu Pandangan struktural atau structural view, pandangan fungsional atau functional view dan pandangan interaksional atau interactional view. Tokoh struktural memandang bahasa sebagai suatu sistem yang secara struktural berkaitan dengan unsur-unsur yang digunakan untuk mengodifikasikan makna.

Menurut pandangan ini biasanya target dan tujuan belajar bahasa adalah penguasaan akan unsur-unsur sistem bahasa (Nurhadi, 1995: 29).

Konsolidasi dari sejumlah kemungkinan definisi bahasa itu menghasilkan definisi gabungan berikut ini:

1. bahasa itu sistematis;
2. bahasa adalah seperangkat simbol manasuka;
3. simbol-simbol itu utamanya adalah vokal, tetapi bisa juga visual;
4. simbol mengonvesionalkan makna yang dirujuk;
5. bahasa dipakai untuk berkomunikasi;
6. Bahasa beroperasi dalam sebuah komunitas atau budaya wicara;
7. Bahasa pada dasarnya untuk manusia, walaupun bisa jadi tak hanya terbatas untuk manusia;
8. Bahasa dikuasai oleh semua orang dalam cara yang sama, bahasa dan pembelajaran bahasa sama-sama mempunyai karakteristik universal

Mengutip pendapat yang dikemukakan Felicia yang menyatakan bahwa pada saat berkomunikasi sehari-hari, salah satu alat yang paling sering digunakan adalah bahasa, baik bahasa lisan maupun bahasa tulis. Begitu dekatnya kita kepada bahasa, terutama bahasa Indonesia, sehingga tidak dirasa perlu untuk mendalami dan mempelajari bahasa Indonesia secara lebih jauh. Akibatnya, sebagai pemakai bahasa, orang Indonesia tidak terampil menggunakan bahasa, suatu kelemahan yang tidak disadari (Suyanto, 2011: 18).

Menurut pendapat dari Ogden & Richard dalam Tarigan (1993: 62), yang mengemukakan adanya lima fungsi bahasa, yaitu:

1. Pelambangan acuan (symbolization of referenceI);
2. Pengekspresian sikap pada penyimak (the expression of attitude to listener);
3. Pengekspresian sikap pada pengacu (the expression of attitude to referent);
4. Penunjang acuan/referensi (support ofreference).

Pada dasarnya, bahasa memiliki fungsi-fungsi sebagai alat untuk mengekspresikan diri, sebagai alat untuk berkomunikasi, sebagai alat untuk mengadakan integrasi dan beradaptasi sosial dalam lingkungan atau situasi tertentu, dan sebagai alat untuk melakukan kontrol sosial.

Berdasarkan pendapat Keraf dalam Finoza (2011: 2) yang menyatakan bahwa dalam literatur bahasa, para ahli umumnya merumuskan fungsi bahasa bagi setiap orang ada empat, yaitu:

1. Sebagai alat komunikasi;
2. Sebagai alat mengekspresikan diri;
3. Sebagai alat berintegrasi dan beradaptasi sosial;
4. Sebagai alat kontrol sosial. Bagi sosiolinguistik konsep bahwa bahasa adalah alat atau berfungsi untuk menyampaikan pikiran dianggap terlalu sempit, sebab dikemukakan Fishman bahwa yang menjadi persoalan sosiolinguistik adalah “Who speak what language to whom, when and to

what end”.

Oleh karena itu, fungsi-fungsi bahasa itu, antara lain, dapat dilihat dari sudut penutur, pendengar, topik, kode dan amanat pembicaraan (Chaer, 2004: 54)

2.2 Algoritma

Algoritma didefinisikan sebagai suatu metode tertentu dalam penyelesaian masalah dengan cara yang mudah dipahami. Di dalam sistem komputer itu sendiri, pengertian algoritma yakni suatu langkah logika yang dibuat dengan memakai software yang telah dirancang oleh pembuat perangkat lunak tersebut. Dalam pembuatan aplikasi dibutuhkan lah penerapan algoritma yang terstruktur agar dapat terselesaikannya aplikasi tersebut dengan lebih sistematis dan teratur. [1]

Algoritma merupakan metode yang efektif yang ditunjukkan pada daftar yang terbatas dari kumpulan perintah yang telah didefinisikan untuk menghitung suatu fungsi. Dalam penyelesaian masalah, ada kriteria-kriteria tertentu pada kondisi awal yang harus dipenuhi sebelum mengeksekusi algoritma.

Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria. Dimulai dari nilai awal, kemudian kumpulan perintah yang pada saat dieksekusi akan memproses kondisi-kondisi yang telah ditetapkan hingga menghasilkan output dan kemudian menentukan kondisi akhir. Pada sistem komputer, algoritma merupakan gambaran langsung dari logika yang dituliskan oleh pembangun perangkat lunak agar lebih efektif dalam pencapaian target

perangkat lunak tersebut, agar dapat memperoleh hasil keluaran dari masukan yang diberikan (terkadang null).

2.3 Sistem

Sistem adalah suatu kumpulan atau himpunan dari unsur , komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Suatu sistem terdiri dari bagian-bagian sistem atau subsistem. Masing-masing subsiste dapat terdiri dari subsistem-subsistem yang lebih lagi atau terdiri dari komponen-komponen pendukung sistem itu sendiri.

Sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen. Dengan pendekatan prosedur , sistem dapat didefinisikan kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Sedangkan dengan pendekatan komponen sistem dapat di definisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya untuk membentuk satu kesatuan untuk mencapai tujuan tertentu. [2]

Sistem juga dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain (Fatta, 2007). Sistem adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan yang kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antarbagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interdependen satu sama lain (Marimin, et-al. 2006).

terdapat banyak jenis dari sistem sehingga dapat diklasifikasikan sebagai berikut sistem abstrak (abstract system), sistem fisik (physical system), sistem tertentu (deterministic system), sistem tak tentu (probabilistic system), sistem tertutup (close system), dan sistem terbuka (open system).

- Sistem tak tentu (probabilistic system), adalah suatu sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
- Sistem abstrak (abstract system), adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.
- Sistem fisik (physical system), adalah sistem yang ada secara fisik.
- Sistem tertentu (deterministic system), adalah sistem yang beroperasi dengan tingkah laku yang tidak dapat diprediksi, interaksi antara bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan.
- Sistem tertutup (close system), adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan.
- Sistem terbuka (open system), adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

2.4 Analisis

Analisis didefinisikan juga sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian komponennya yang bertujuan untuk mengidentifikasi dan mengevaluasi kesempatan, permasalahan, hambatan yang terjadi dan kebutuhan yang diharapkan dapat diusulkan perbaikannya. Analisis sangat berguna untuk mengetahui function-function dari proses aktivitas utama yang dilakukan pada sistem yang sedang berjalan sehingga akan lebih mempermudah bagi programmer dalam mengembangkan aplikasi.

2.5 Aplikasi

Ada banyak pengertian mengenai aplikasi menurut beberapa ahli yaitu Menurut Supriyanto (2005:2) adalah program yang memiliki aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu, kemudian Menurut Kamus Besar Bahasa Indonesia adalah penerapan dari rancang

Menurut Kadir (2008:3) program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- 1) Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.

- 2) Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Dari kedua pengertian di atas dapat disimpulkan bahwa aplikasi adalah sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau hardware komputer yang digunakan oleh manusia dalam menjalankan program aplikasi, dengan demikian bisa membantu manusia untuk memberikan solusi dari apa yang diinginkan.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (Application suite). Contohnya adalah Microsoft Office dan OpenOffice.org. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap hari. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki beberapa kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Umumnya aplikasi-aplikasi tersebut memiliki kemampuan untuk saling berinteraksi sehingga menguntungkan pengguna itu sendiri. Contohnya, suatu lembar kerja dapat dimasukkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah

2.6 bahasa pemrograman

Bahasa pemrograman merupakan sebuah perintah atau instruksi untuk memerintah komputer agar menjalankan fungsi tertentu dengan hasil yang standar, rendah, maupun high. Saat ini yang paling banyak mengalami perkembangan adalah bahasa pemrograman tingkat tinggi. Tercatat sejak kemunculan bahasa C yang dibuat oleh Dennis Ritchie, yang kemudian menjadi pondasi dari beberapa bahasa pemrograman seperti Java (James Gosling), C++ (Bjarne Stroustrup), PHP (Rasmus Lerdorf), dan sebagainya. Untuk berkomunikasi pada sebuah komputer/mesin dibutuhkan adanya bahasa pemrograman sehingga kita dapat memasukkan instruksi-instruksi kedalam komputer/mesin melalui bahasa pemrograman. Bahasa pemrograman juga akan mempermudah programmer dalam pembuatan aplikasi sehingga aplikasi menjadi lebih menarik dan mudah untuk digunakan.. [4]

Adapun konsep-konsep dasar dalam pemrograman adalah sebagai berikut :

9. Simulasi , sensibilitas terhadap masalah dan kemungkinan solusi. Kegiatan dilakukan di perusahaan, melalui permainan. Contoh : Mengurutkan nama setiap karyawan yang ada di kantor dimulai dari huruf A sampai Z. Hal ini dapat dilakukan secara komputerisasi maupun manual.
10. Analisis masalah secara lebih formal dan membuat spesifikasi dan algoritma dalam notasi yang ditetapkan. Mahasiswa harus menuliskan solusi algoritmiknya dalam notasi standar di kelas. Penulisan notasi algoritmik bertujuan untuk menyeragamkan pemahaman tentang algoritma program yang terbebas dari sintak (aturan) penulisan bahasa program

11. Menulis program, yaitu menterjemahkan notasi algoritmik ke dalam sintak bahasa program.
12. Debugging dan menguji coba program. Hal ini bertujuan untuk mendapatkan program yang benar. Program dikatakan benar jika terbebas dari salah logik dan sintak bahasa. Secara ideal mahasiswa hanya diberi 16 kesempatan untuk me-run program sebanyak 2 kali : pertama untuk membersihkan program dari kesalahan sintak dan kedua untuk mendapatkan program benar. Pada tahap ini diharapkan tidak terjadi kesalahan logik jika analisa benar
13. Mengamati peristiwa eksekusi, perlu dilakukan untuk meningkatkan kepercayaan bahwa jika analisa benar maka sisa pekerjaan menjadi mudah. Pada pemrograman prosedural, aspek ini penting untuk memahami fenomena eksekusi dan perubahan nilai suatu struktur data.
14. Membaca program : orang akan dapat menulis dengan baik kalau sering membaca. Hal ini juga berlaku dalam memprogram. Kegiatan yang dapat dilakukan di kelas adalah dengan saling tukar menukar teks algoritma, dan saling mengkritik algoritma teman. Mahasiswa harus berlatih sendiri pada kegiatan belajar bersama.
15. Membuktikan kebenaran program secara formal , satu-satunya hal yang menjamin kebenaran, tetapi kontradiktif dan sulit diterapkan dalam kehidupan sehari-hari. Program yang hanya lima baris pembuktiannya bisa sehalaman, sehingga seringkali tidak pernah diterapkan dalam aplikasi nyata. Aktifitas ini dicakup dalam matakuliah Analisis Algoritma

2.5.1 PHP

PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan". Php disebut juga pemrograman Server Side Programming, hal ini dikarenakan seluruh prosesnya dijalankan pada server. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan open source yaitu pengguna dapat mengembangkan kode-kode fungsi sesuai kebutuhannya. Pada awalnya PHP dikenal dengan singkatan Personal Home Page. Karena server tersebut diperuntukan untuk website pribadi. Tetapi untuk saat ini PHP sudah berubah menjadi bahasa pemrograman yang sangat populer yang tentunya digunakan untuk website terkenal seperti Wikipedia, wordpress, Joomla dan lain-lain. Untuk saat ini PHP dikenal dengan singkatan Hypertext Preprocessor sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri. Bahasa pemrograman PHP banyak digunakan karena sifatnya yang open source yaitu dapat digunakan secara gratis. Fungsi bahasa pemrograman PHP itu sendiri adalah web tersebut digunakan untuk dapat menyesuaikan tampilan berupa sebuah konten yang sesuai dengan situasi. Web yang dinamis juga digunakan untuk menyimpan data ke database dengan memproses form dan juga nantinya dapat mengubah tampilan website tersebut sesuai inputan dari seorang user.

2.5.2 My SQL

SQL (Structured Query Language) merupakan bahasa basis data yang memungkinkan pengguna untuk membuat basis data dan struktur relasional, melakukan tugas dasar manajemen manajemen data, seperti insertion, modification dan deletion data dari relasi serta melakukan query sederhana maupun query yang kompleks. Pada aplikasi data/informasi yang di input akan disimpan secara otomatis pada database yang dihubungkan melalui My SQL.

Sebagai database yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL :

- 1) Portability MySQL dapat berjalan stabil pada berbagai sistem operasi di antaranya adalah seperti Windows, Linux, FreeBSD, Mac OS X server, Solaris, Amiga, HP-UX dan masih banyak lagi.
- 2) Open Source MySQL didistribusikan secara open source (gratis), di bawah lisensi GPL.
- 3) Multiuser MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan.

- 4) Performance Tuning MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- 5) Column Types MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, year, set serta enum.
- 6) Command dan Function MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.
- 7) Security MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan user dengan system perizinan yang mendetail serta password terencripsi.
- 8) Stability dan Limits MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu table serta 5 miliar baris. Selain itu, batas indeks yang dapat di tampung mencapai 32 indeks pada tiap tabelnya.
- 9) Connectivity MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix soket (Unix), atau Named Pipes (NT).

- 10) Localisation MySQL dapat mendeteksi pesan kesalahan (error code) pada client dengan menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termasuk di dalamnya.
- 11) Interface MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
- 12) Client dan Tools MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertai petunjuk online.
- 13) Struktur Tabel MySQL memiliki struktur table yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

2.7 database

Database merupakan kumpulan berbagai data serta informasi tertentu yang tersimpan serta tersusun secara sistematis dalam komputer sehingga dapat diperiksa maupun diolah serta di manipulasi dengan bantuan program komputer agar informasi dari basis data tersebut. Database atau sering disebut juga basis data memiliki berbagai manfaat seperti mengurangi terjadinya redundansi atau pengulangan, integritas data lebih terjaga, dan mempermudah dalam berbagi data. Database sangat dibutuhkan dalam pembuatan suatu aplikasi dikarenakan basis data memberikan kemudahan bagi user dalam membagikan data kepada pengguna. [5]

Objek-objek dalam sebuah database adalah sebagai berikut :

- 1) Database Database berisi berbagai objek yang digunakan untuk mewakili, menyimpan data, dan mengakses data.
- 2) Table Objek yang berisi tipe-tipe data dan data mentah
- 3) Kolom Sebuah tabel berisi kolom-kolom untuk menampung data. Kolom mempunyai sebuah tipe dan nama yang unik.
- 4) Tipe data Sebuah kolom mempunyai sebuah tipe data. Tipe-tipe yang dapat dipilih adalah karakter, numeric, tanggal, Boolean dan lain-lain.
- 5) Stored procedure Merupakan perintah-perintah SQL yang membentuk makro. Dengan menjalankan stored procedure berarti menjalankan perintah-perintah SQL di dalam sebuah procedure.
- 6) Trigger Stored procedure yang diaktifkan pada saat data ditambahkan, diubah, atau dihapus dari database. Trigger dipakai untuk menjamin aturan integrasi di dalam

database.

- 7) Rule Diberlakukan pada kolom sehingga data yang dimasukan harus sesuai dengan aturan.
- 8) Primary key Menjamin setiap baris data unik, dapat dibedakan dari data yang lain.
- 9) Foreign key Kolom-kolom yang mengacu primary key pada table lain. Primary key dan foreign key dipakai untuk menghubungkan sebuah data dengan tabel lain.
- 10) Konstrain Mekanisme integritas data yang berbasis server dan diimplementasikan oleh sistem. 27
- 11) Default Dinyatakan pada field (kolom) sehingga jika kolom tersebut tidak diisi data, maka diisi dengan nilai default.
- 12) View Query yang memakai beberapa tabel, dan disimpan di dalam database. View dapat memilih beberapa kolom dari sebuah tabel atau menghubungkan beberapa tabel. View dapat dipakai untuk menjaga keamanan data.
- 13) Index Membantu mengorganisasikan data sehingga query menjadi lebih cepat.
- 14) Fungsi Kumpulan perintah yang mengandung input atau tidak menggunakan input baik satu atau lebih dari satu dan mengeluarkan nilai baik berupa skalar maupun tabular (berbentuk tabel).

2.8 Google API

API merupakan akronim untuk Application Programming Interface, yang merupakan perantara Software (Perangkat Lunak) yang memungkinkan 2 (Dua) Application (Ap-likasi)

untuk saling berbicara. Application Programming Interface atau API berfungsi menyediakan Function dan juga Perintah dengan bahasa yang lebih terstruktur dan lebih mudah bagi dipahami bagi Programmer jika dibandingkan dengan System Calls, hal ini adalah merupakan hal yang penting dalam aspek Editing dan pengembangan- nya, sehingga Programmer mampu mengembangkan Sistem tersebut dengan mudah.

API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang kita buat dengan fungsi-fungsi yang dikerjakan. Misalnya dalam hal ini Google API berarti kode program (yang disederhanakan) yang dapat kita tambahkan pada aplikasi atau web kita untuk mengakses/menjalankan/memanfaatkan fungsi atau fitur yang disediakan Google. Misalnya saja kita bisa menambahkan fitur Google Map pada website kita.

Google API dapat dipelajari langsung melalui Google Code. Melalui Google Code kita dapat belajar tentang Google API dan dapat mengimplementasikan pada aplikasi web atau website yang kita kembangkan.

Ada banyak API yang disediakan oleh Google, beberapa diantaranya adalah:

- Language API: untuk memanfaatkan fitur translation yang dimiliki Google.
- Earth API: memanfaatkan fitur yang ada pada Google Earth
- Javascript API
- Maps API: memanfaatkan fitur yang ada pada Google Maps
- Search API: memanfaatkan fitur pencarian pada Google Search
- Visualization API: membuat grafik maupun chart dengan Google API

- YouTube API: memanfaatkan fitur yang ada pada YouTube misalnya untuk pencarian video

Salah satu cara mudah mempelajari Google API adalah dengan memanfaatkan Google AJAX APIs Playground. AJAX APIs playground adalah sebuah situs yang disediakan oleh Google bagi kita untuk mencoba secara langsung sejumlah Google API yang berbasis AJAX (Asynchronous Javascript and XML). Karena berbasis AJAX maka tentunya semua kode program dalam sintaks Javascript yang bisa kita lihat, kopi dan paste secara langsung untuk digunakan pada website kita.

Dengan menggunakan Google AJAX API, kita bisa mengintegrasikan data pada website kita dengan API yang disediakan oleh Google.

2.9 BPMN

tandar notasi grafis yang mendeskripsikan langkah logis dari proses bisnis. BPMN diterima sebagai standar pemodelan proses bisnis baik secara nasional maupun internasional. Secara nasional, BPMN dimuat pada peraturan Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 12 Tahun 2011. BPMN digunakan

untuk sebagai acuan menyusun Standard Operating Procedures (SOP) bagi kementerian atau lembaga pemerintah daerah pada peraturan tersebut. Secara internasional, BPMN distandarkan melalui ISO/IEC 19510:2013 Information Technology – Object Management Group Business Process Model and Notation.

2.10 UML

A.Suhendar dan Hariman Gunadi menuliskan dalam bukunya (Suhendar dan Gunadi, 2002) bahwa Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam OOAD (Object Oriented Analysis and Desain) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasi artifact yang terdapat dalam software.

UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO (object oriented) yang telah ada sebelumnya, yaitu Booch, OMT, dan OOSE. UML merupakan kesatuan dari ketiga metode pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani oleh ketiga metode tersebut.

Dimana tujuan dari UML ini adalah:

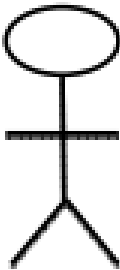





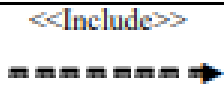
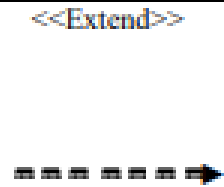
- Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum,
- Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa, dan

- Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

2.9.1 Use Case Diagram

Use case menggambarkan external view dari sistem yang akan kita buat modelnya (Prabowo Pudjo Widodo, 2011) Model use case dapat dijabarkan dalam diagram use case, tetapi perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. (Pooley, 2003:15). Use case harus mampu menggambarkan urutan aktor yang menghasilkan nilai terukur (Prabowo Pudjo Widodo, 2011). [6]






Berikut pengertian use case diagram menurut Satzinger (2011 : 20) “Use Case Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan”.

Simbol	Penjelasan
	<i>Actor</i> Menspesifikasikan seperangkat peranan yang user sistem dapat diperankan ketika berinteraksi dengan <i>use case</i> .
	<i>Association</i> Menggambarkan interaksi antara <i>actor</i> dan <i>use case</i> .
	<i>Generalization</i> Relasi antar <i>use case</i> , dimana salah satunya dalam bentuk yang lebih umum dari yang lain.
	<i>Use Case</i> Sebuah deskripsi dari seperangkat aksi-aksi berurutan yang ditampilkan pada sebuah sistem.
	<i>System</i> Tempat seluruh aktivitas-aktivitas sistem yang sedang berjalan.
	<i>Dependancy</i> Untuk menggambarkan ketergantungan sebuah <i>use case</i> dengan <i>use case</i> lainnya.
	<i>Include</i> Menggambarkan bahwa keseluruhan dari sebuah <i>use case</i> merupakan fungsionalitas <i>use case</i> lainnya.
	<i>Extend</i> Menggambarkan hubungan antar <i>use case</i> dimana bahwa sebuah <i>use case</i> merupakan fungsionalitas <i>use case</i> lainnya apabila kondisi tertentu terpenuhi.

tabel 3.6 Tipe relasi pada use case diagram

2.9.2 Activity Diagram

Diagram activity menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. Activity diagram juga dapat menggambarkan proses lebih dari satu aksi salam waktu bersamaan. “Diagram activity adalah aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas” (Haviluddin, 2011). [7]

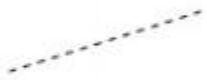
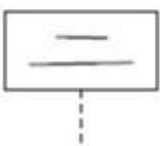


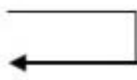
Simbol	Penjelasan
	<i>Initial State</i> Mempresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i> .
	<i>Action State</i> Sebuah <i>state</i> yang menggambarkan eksekusi dari aksi <i>atomic</i> .
	<i>Transitton Between Activities</i> Mengidentifikasi bahwa suatu objek dari state pertama akan menampilkan aksi-aksi tertentu dan memasuki state kedua ketika peristiwa terjadi pergerakan dari aksi ke aksi lainnya.
	<i>Decision Point</i> Menentukan kapan alur dalam aktivitas menjadi bercabang.
	<i>Final State</i> Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam <i>activity diagram</i> .

tabel 3.5 Tipe relasi activity diagram

2.9.3 Sequence Diagram

“Sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram.” (Haviluddin, 2011). [8]

Tabel 3.6 simbol simblo pada sequence diagram

SIMBOL	NAMA	KETERANGAN
	Anchor Note to item	Melekatkan catatan pada use case atau aktor tertentu dalam diagram.
	Object	Menambahkan objek baru pada diagram.
	Object Message	Menggambarkan pesan antar dua objek.
	Return Message	Menggambarkan pengembalian diri pemanggilan prosedur.
	Message to self	Menggambarkan pesan yang menuju dirinya sendiri.

2.9.4 Class Diagram

Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek (Whitten, 2004:410). Class memiliki tiga area pokok yaitu :

- Nama, kelas harus mempunyai sebuah nama.
- Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.

- Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya. [9]

2.11 BlackBox Testing

Dalam jurnal yang dituliskan oleh Putu Sudharyana, Bektı Cahyo Hidaryanto, Feby Artwodini Muqtadiroh (Sudharyana, et-al. 2012) bahwa Black box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang berten- tangan dengan struktur internal atau kerja (lihat pengujian white-box). Pengetahuan khusus dari kode aplikasi/struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Uji coba black box berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

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

Black Box Testing terfokus pada spesifikasi fungsional dari perangkat lunak.

2.12 WhiteBox Testing

Menurut Pressman (2010:588) “Pengujian white box (Pengujian Kotak Putih), terkadang disebut juga pengujian kotak kaca (glass box testing), merupakan sebuah filosofi perencanaan test case (uji kasus) yang menggunakan struktur control yang dijelaskan sebagai bagian dari perancangan perangkat komponen untuk menghasilkan test case”.

Menurut Pressman (2010:588) Dengan menggunakan metode pengujian kotak putih, anda dapat memperoleh test case yang :

1. Menjamin bahwa semua jalur independen dalam sebuah modul telah dieksekusi setidaknya satu kali.
2. Melakukan semua keputusan logis pada sisi benar dan yang salah.
3. Melaksanakan semua loop (putaran) pada batas mereka dan dalam batas-batas operasional mereka.
4. Melakukan struktur data internal untuk memastikan kesahihannya.

Kelebihan dan kelemahan pengujian white box testing dapat diuraikakn sebagai berikut :

1. Kelebihan White Box Testing :
 - a. Kesalahan Logika Digunakan pada sintaks “If “ dan pengulangan. Dimana white box testing akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.
 - b. Ketidak Sesuaian Asumsi Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
 - c. Kesalahan Ketik Mendeteksi bahasa pemrograman yang bersifat case sensitive.

2. Kelemahan White Box Testing: Untuk perangkat lunak yang tergolong besar, white box testing dianggap sebagai strategi yang tergolong boros, karena akan melibatkan sumber daya yang besar untuk melakukannya.

Dalam pengujian white box testing ada beberapa langkah dalam menjalankan pengujiannya :

1. Mendefinisikan semua alur logika
2. Membangun kasus untuk digunakan dalam pengujian
3. Melakukan pengujian

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

3.1.1 Analisis Sistem Yang Sedang Berjalan (*Current System*)

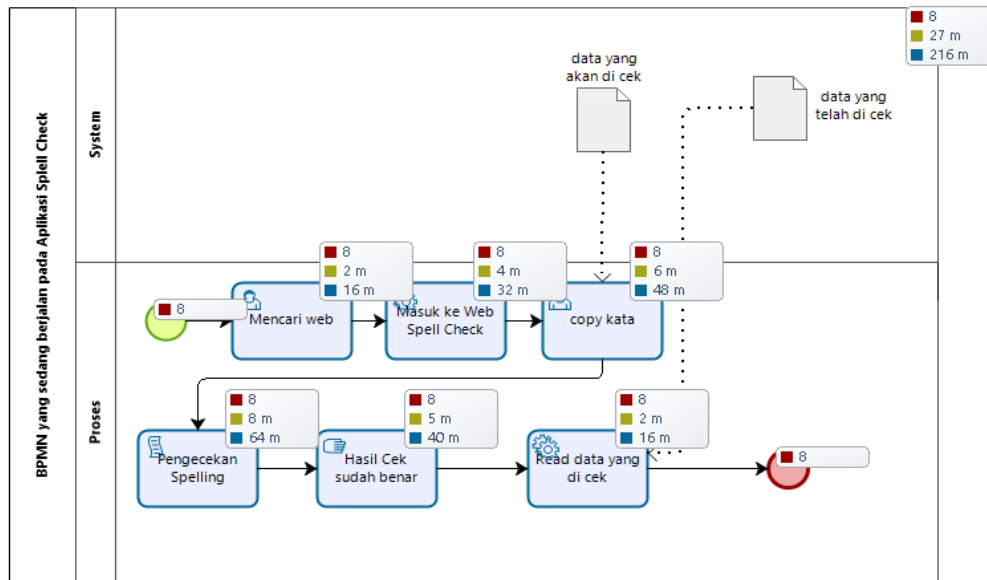
Dari hasil analisis yang telah dilakukan maka alur yang sedang berjalan pada Aplikasi Spell Checker adalah sebagai berikut:

3.1.1.1 Analisis BPMN Yang Sedang Berjalan

System yang sedang berjalan saat ini yaitu terdiri dari proses *copy* kata dan proses melihat hasil cek Spelling.

Proses pertama seorang *User* akan mencari web untuk melakukan *Spelling check* apabila *User* telah menemukan web nya lalu *User* akan masuk kedalam web tersebut. Setelah masuk ke web selanjutnya *User* akan men *copy* kata yang nantinya akan di cek. Selanjutnya *System* akan bekerja yaitu akan mencek kata tersebut. Apabila prosesnya telah selesai maka *User* akan dapat melihat hasil cek kata.

Adapun *Bizagi Modeler* pada Aplikasi Spell Check adalah sebagai berikut:



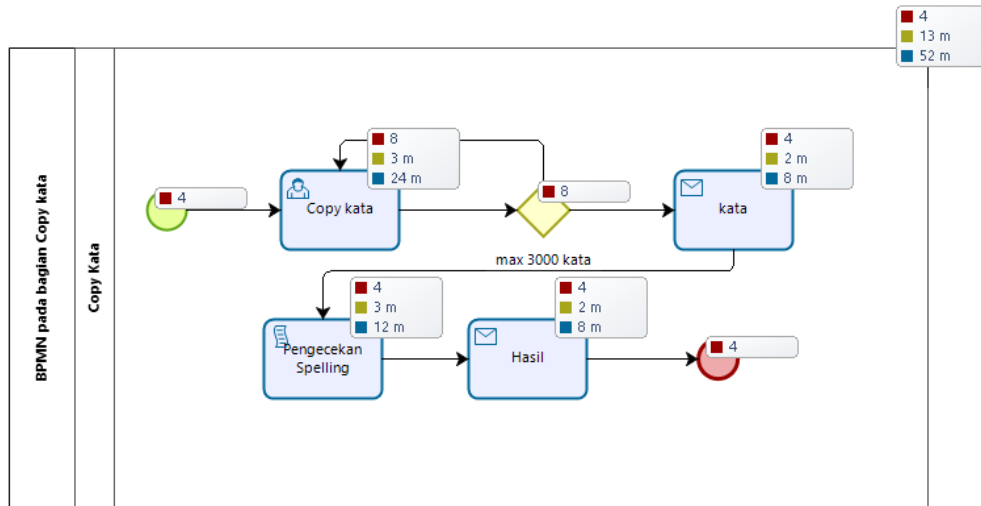
Gambar 3.1 Bizagi Modeler sistem berjalan pada Aplikasi Spell Check

<i>Name</i>	<i>Type</i>	<i>Total Time</i>
Bpmn Yang Sedang berjalan Pada Aplikasi Cek Karya Ilmiah	<i>Process</i>	3 h 36 m
<i>None Start</i>	<i>Start event</i>	-
Mencari web	<i>Task</i>	16m
Masuk ke web Spell Check	<i>Task</i>	32m
Copy kata	<i>Task</i>	48m
Pengecekan Spelling	<i>Task</i>	1h 4m
Hasil cek sudah benar	<i>Task</i>	40m
Read data yang dicek	<i>Task</i>	16m
<i>None End</i>	<i>End Event</i>	-

abel 3.1 Hasil simulasi proses sistem berjalan pada Aplikasi Spell Check

A. Analisis BPMN Yang Sedang Berjalan Pada Proses Copy Kata

User akan mencari kata yang akan di cek lalu *dicopy* pada text box yang tersedia , apabila melebihi batas minimum yaitu 3000 kata maka *User* akan kembali lagi untuk mengurangi kata yang di cek agar tak melebihi batas minimum. Selanjutnya apabila sudah ada kata dalam teks box maka proses selanjutnya yaitu aplikasi akan bekerja untuk melakukan pengecekan *Spelling*. Setelah proses selesai maka akan muncul hasil dari cek *Spelling*. Adapun *Bizagi Modeler* Aplikasi Spell Check sebagai berikut :



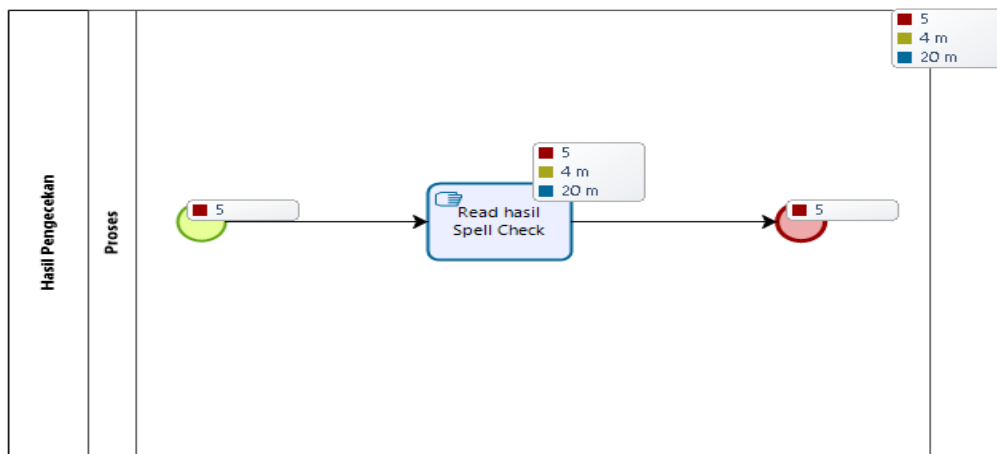
Gambar 3.2 Bizagi Modeler proses Copy kata sistem yang sedang Berjalan

Tabel 3.2 Hasil simulasi proses sistem berjalan pada Aplikasi Cek Karya Ilmiah

Name	Type	Total Time
Copy Kata	Prosess	52m
None Start	Start event	-
Max 3000 kata	Gateway	
None End	End Event	-
Copy Kata	Task	24m
Kata	Task	8m
Pengecekan Spelling	Task	12m
Hasil	Task	8m

B. Analisis Yang Sedang Berjalan Pada Proses Hasil Cek Karya Ilmiah

C.



Gambar 3.3 Bizagi Modeler proses Hasil Cek Karya Ilmiah sistem yang sedang

Tabel 3.3 Hasil simulasi proses hasil cek karya ilmiah sistem berjalan

Name	Type	Total Time
Hasil Pengecekan	Process	4m
None Start	Start event	-
None End	End Event	-
Read hasil Spell check	Task	4m

3.1.1.2 Analisis Dokumen yang Digunakan

Dari hasil analisis yang dilakukan, dokumen yang digunakan diantaranya oleh Aplikasi *Spell check* diantaranya dokumen proses *Copy* data dan dokumen proses hasil cek *Spelling*. Dokumen yang dimaksud adalah sebagai berikut :

Tabel 3.4 Dokumen proses Copy Kata

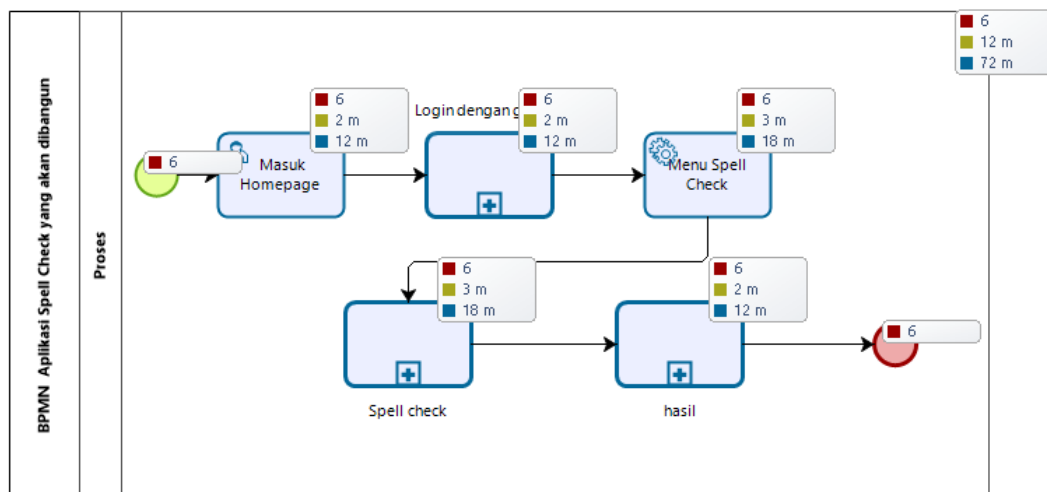
Dibuat oleh	<i>System</i>
Dibuat untuk	<i>User</i>
Isi	Proses <i>Copy</i> data pada <i>Spell check</i>
Frekuensi	Dibuat sesuai kebutuhan <i>User</i>
Tujuan	Langkah-langkah dalam proses <i>Copy</i> kata

Tabel 3.5 Dokumen proses hasil cek Spelling

Dibuat oleh	<i>System</i>
Dibuat untuk	<i>User</i>
Isi	Proses hasil dari cek Spelling
Frekuensi	Dibuat sesuai kebutuhan <i>User</i>
Tujuan	proses hasil setelah <i>System</i> bekerja untuk cek <i>Spelling</i>

3.1.2 Analisis Sistem yang akan Dibangun

Analisis kebutuhan yang dimaksud di sini berupa analisis *Bizagi Modeler* mengenai sistem yang akan dibangun meliputi sub proses *log in*, dan sub proses hasil dari *Spelling check*. Adapun *Bizagi Modeler* yang akan dibangun adalah sebagai berikut:



Gambar 3.4 Bizagi Modeler sistem yang akan dibangun

Tabel 3.6 Hasil simulasi proses sistem yang akan dibangun

<i>Name</i>	<i>Type</i>	<i>Total Time</i>
Bpmn Aplikasi Spell Check yang akan dibangun	<i>Process</i>	1h 12m
<i>None Start</i>	<i>Start event</i>	
<i>None End</i>	<i>End Event</i>	
Masuk ke Home Page	<i>Task</i>	12m
Menu spell check	<i>Task</i>	18m
Login dengan google	<i>Task</i>	12m
<i>Spell check</i>	<i>Task</i>	18m
Hasil	<i>Task</i>	12m

3.1.2.3 Analisis BPMN yang akan Dibangun

A. Analisis Sistem yang akan Dibangun Pada Sub Proses *Log in*

Di sini *User* akan masuk kedalam halaman *log in*, apabila *User* tersebut telah mempunyai akun maka dia akan langsung masuk ke halaman utama cek karya ilmiah. Apabila *User* tersebut belum memiliki akun maka *User* akan membuat akun terlebih dahulu ,apabila telah memiliki akun maka *User* akan bisa masuk ke halaman utama aplikasi cek karya ilmiah.

belomm

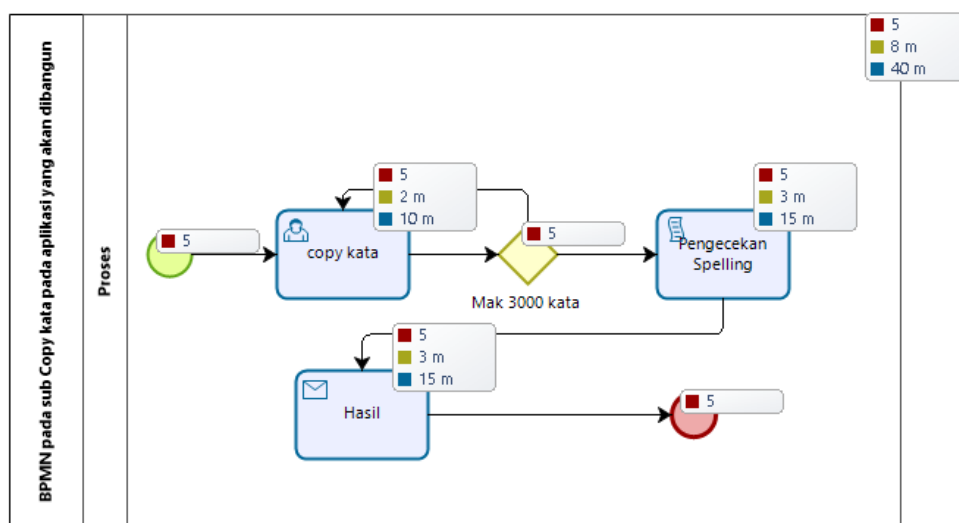
Gambar 3.5 Bizagi Modeler sub proses log in aplikasi yang akan dibangun

Tabel 3.7 Hasil simulasi sub proses log in aplikasi yang akan dibangun

Name	Type	Total Time
Proses Login	Start event	
Buat Akun Google	Task	
Punya akun Google	Task	
None Start	Start event	
Halaman Utama	Task	
None End	End Event	
Exclusive Gateway	Gateway	

B. Analisis Sistem yang akan Dibangun Pada Sub Proses Copy Data

User akan mencari data yang akan dicek , apabila belum maka akan kembali ke proses cari data, setelah data ditemukan lalu dapat melakukan proses pengecekan karya ilmiah. Setelah proses pengecekan karya ilmiah selesai, maka akan tampil hasil dari



proses tersebut.

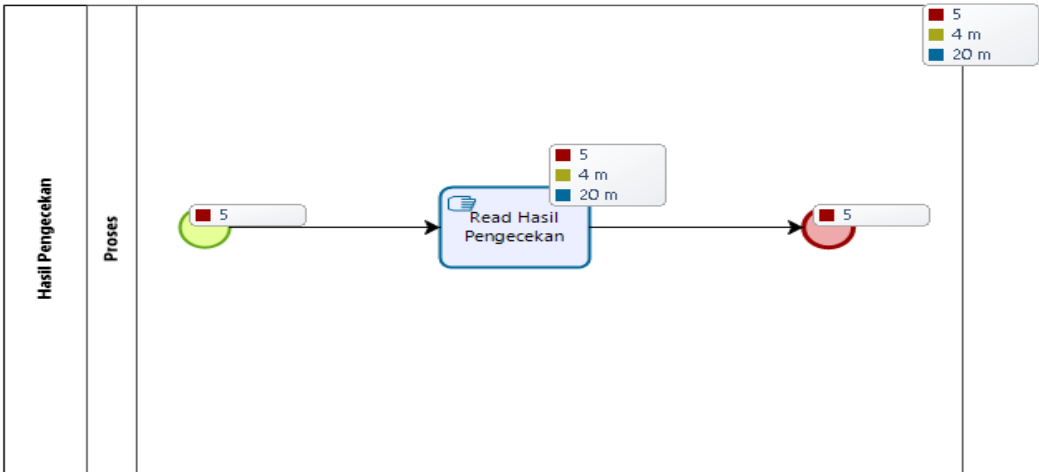
Gambar 3.6 Bizagi Modeler sub proses Copy Kata pada aplikasi yang akan dibangun

Tabel 3.8 Hasil simulasi sub proses Copy kata pada yang akan dibangun

Name	Type	Total Time
BPMN pada sub Copy kata pada aplikasi yang akan dibangun	Process	40m
None Start	End Even	
None End	Start Even	
Max 30000 kata	Task	
Hasil	Task	15m
Pengecekan Spelling	Task	15m
Copy Kata	Task	10m

C. Analisis Sistem yang akan Dibangun Pada Sub Proses Hasil Pengecekan

User akan melihat hasil dari cek karya ilmiah yang telah selesai, di sana User juga bisa



untuk men *download* data tersebut. Adapun *Bizagi Modeler* proses hasil sebagai berikut:

Gambar 3.7 Bizagi Modeler sub proses Hasil aplikasi yang akan dibangun

Tabel 3.9 Hasil simulasi sub proses hasil aplikasi yang akan dibangun

<i>Name</i>	<i>Type</i>	<i>Total Time</i>
Hasil Pengecekan	<i>Process</i>	4m
Read hasil pengecekan	<i>Task</i>	4m
<i>None End</i>	<i>End Event</i>	
<i>None Start</i>	<i>Start event</i>	

3.1.2.2 Kebutuhan Fungsional (*Functional Requirements*)

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan kebutuhan sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan fungsional yang akan dibuat yaitu:

1. *Login User*
2. *Kelola data User*

Setiap proses memiliki representasi masing-masing pada sebuah tabel atau data yang terdapat pada *database* yang telah dirancang sebelumnya. Dan setiap proses berhubungan langsung dengan entitas atau *User*.

3.1.2.3 Kebutuhan Non-Fungsional (*Non- Functional Requirement*)

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat lunak/*software*, analisis perangkat keras/*hardware*, analisis pengguna/*User*. Adapun kebutuhan fungsional yang akan dibuat adalah sebagai berikut :

a. Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan dalam aplikasi *Spell check* adalah sebagai berikut:

Tabel 3.10 Software Requirement:

No.	Tools/Software	Fungsi
1.	<i>Windows 7</i>	Sistem Operasi
2.	<i>PHP dan CI</i>	Bahasa Pemrograman
3.	<i>Xampp 1.7.3</i>	Server Basis Data
4.	<i>MySQL 5.0.11</i>	DBMS
5.	<i>Sublime Text, Bizagi Modeler, Star UML</i>	Perangkat Lunak
6.	<i>Framework code igniter</i>	Aplikasi Pelengkap

b. Perangkat Keras yang Digunakan

Perangkat keras yang digunakan dalam aplikasi *Spell check* adalah sebagai berikut:

Tabel 3.11 Hardware Requirement

	nama Perangkat	Spesifikasi	Keterangan
	tor		cepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor computer
			system yang akan digunakan
		HDD; (min)	untuk menyimpan data aplikasi yang dibuat
	tur jaringan		alogikan sebagai alur proses dari titik awal proses sampai pada akhir proses.

c. Analisis Pengguna/User

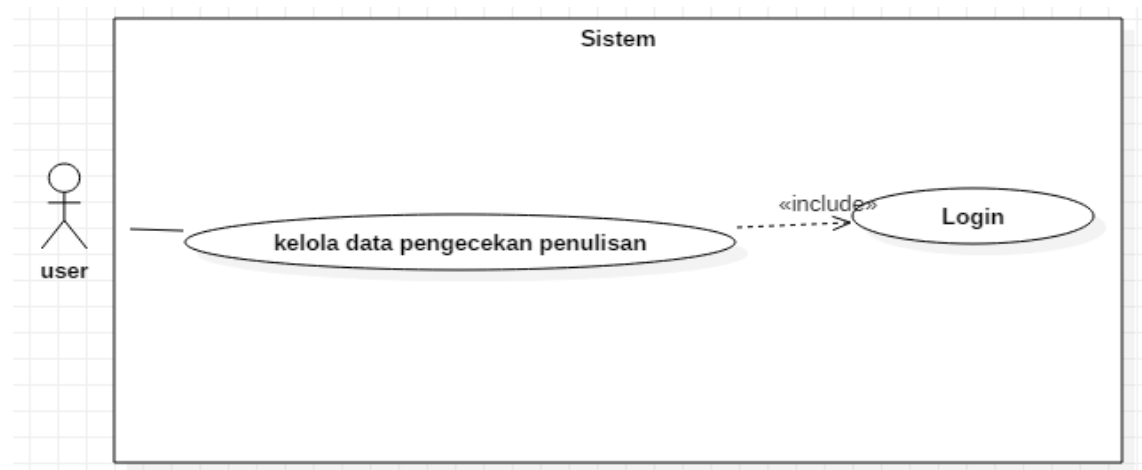
Aplikasi yang akan dibuat ini digunakan dalam lingkungan mahasiswa, khususnya pada mahasiswa Politeknik Pos Indonesia. Yang melibatkan *User* sebagai penggunanya.

3.2 Perancangan

Berikut ini adalah suatu *Gambar 3.* analisis pada perancangan dan pembuatan Aplikasi pengecekan karya ilmiah menggunakan UML, yang terdiri dari *Use Case*, *Class Diagram*, *Sequence Diagram*, *Collaboration Diagram*, *Activity Diagram*, *State chart Diagram*, *Component Diagram*, *Deployment Diagram*, *Object Diagram* dan *Struktur Diagram*.

3.2.1 Use Case Diagram

Berikut ini merupakan *Use Case Diagram* yang menjelaskan proses interaksi antara aktor dengan *system* yang telah dirancang. Di sini *User* untuk melakukan pengecekan penulisan harus log in terlebih dahulu kedalam aplikasi tersebut.



Gambar 3.8 Use Case Diagram Pengecekan

3.2.1.1 Skenario Use Case

3.2.1.1.1 Definisi Aktor

Pada bagian ini dijelaskan aktor yang terlibat dalam aplikasi *spell check* Ilmiah sebagai berikut :

Tabel 3.12 Definisi Aktor

No	Aktor	Deskripsi
1.	<i>User</i>	a. <i>Login</i> Aplikasi b. Melakukan pengecekan penulisan

3.2.1.1.2 Definisi Use Case

Use case merupakan teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi antara para pengguna sistem dengan sistem itu sendiri, dengan memberikan sebuah narasi tentang bagaimana sistem tersebut digunakan.

Tabel 3.13 Definisi Use Case Login

No.	Use Case	Deskripsi
1.	<i>Login</i>	<i>User</i>
2.	Pengecekan <i>Spelling</i>	a. Menampilkan <i>form</i> Pengecekan <i>Spelling</i> b. Men <i>Copy</i> kata Pengecekan <i>Spelling</i>

3.2.1.1.3 Skenario Use Case

1. *Login*

Tabel 3.14 skenario Use Case Login

Identifikasi	
Nama	<i>Login</i>

Tujuan	Melakukan <i>Login</i> untuk masuk halaman utama
Deskripsi	<i>Login</i> dengan memasukan <i>Username</i> dan Password
Aktor	<i>User</i>
Skenario	
Kondisi Awal	Menampilkan <i>form Login</i>
Kondisi Akhir	Menampilkan <i>form</i> halaman utama
Aksi Aktor	Reaksi Sistem
1. Aktor meminta fasilitas <i>login</i> kepada sistem	2. Menampilkan <i>form login</i>
3. Memasukan <i>Username</i> dan Password Membuat akun baru	4. Melakukan proses validasi jika salah keduanya muncul pesan “ <i>Username</i> dan “ <i>Password</i> ” yang anda masukan tidak benar”.
	5. Jika benar maka akan ditampilkan <i>form</i> halaman utama

2. Pengecekan Penulisan

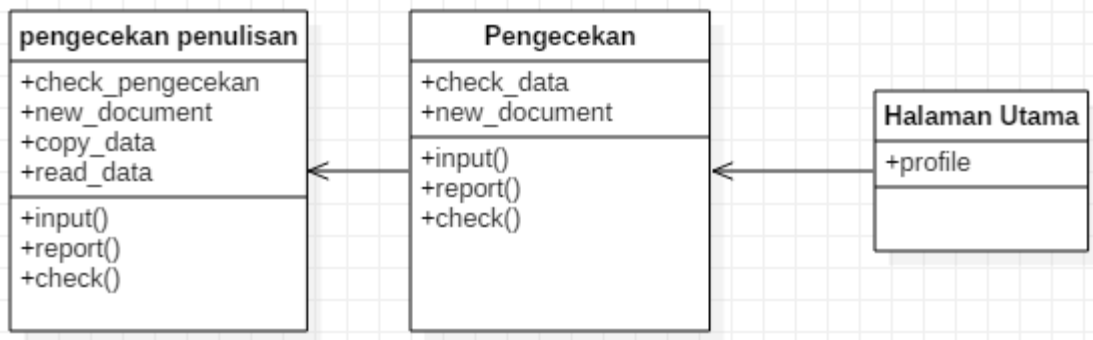
Tabel 3.15 Use case Pengecekan *Speeling*

Identifikasi	
Nama	Pengecekan Ejaan
Tujuan	Dapat melakukan proses Pengecekan <i>Spelling</i>
Deskripsi	Menu yang digunakan untuk melakukan proses Pengecekan Data Ilmiah
Aktor	<i>User</i>
Aksi Aktor	Reaksi System

1. Aktor meminta fasilitas Pengecekan Penulisan kepada <i>System</i>	2. Menampilkan halaman pengecekan <i>Spelling</i>
--	---

3.2.2 *Class* Diagram

Berikut ini merupakan *Class Diagram* yang menjelaskan proses *class* apa saja yang terdapat dalam *system* yang telah dibuat.



Gambar 3.9 Class Diagram Aplikasi Pengecekan Spelling

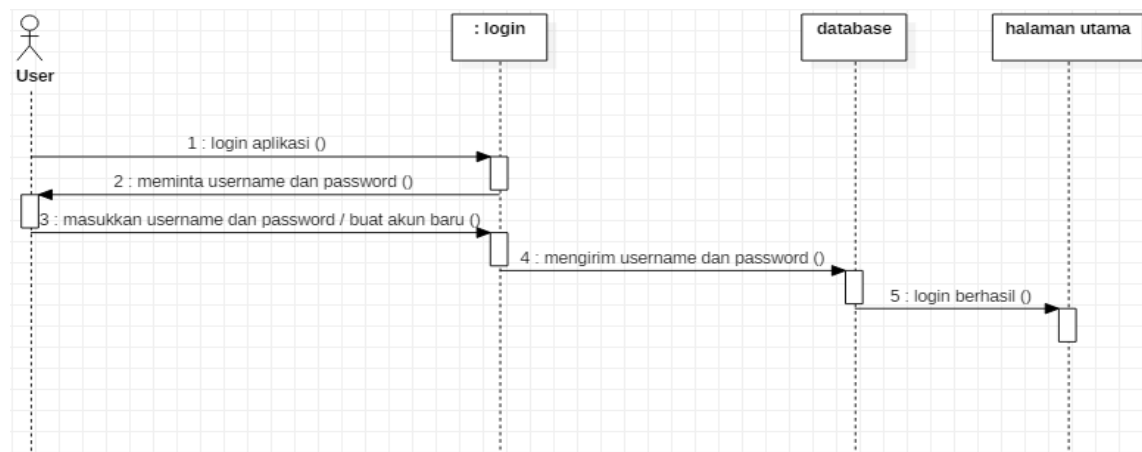
Pada *Class Diagram* aplikasi Pengecekan Penulisan Data Ilmiah terdapat *class-class* diantaranya, *class Login*, *User*, dan pengecekan penulisan.

3.2.3 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu.

3.2.3.1 Sequence Diagram Login User

Berikut ini merupakan *Sequence diagram login* yang menjelaskan *proses login User*.



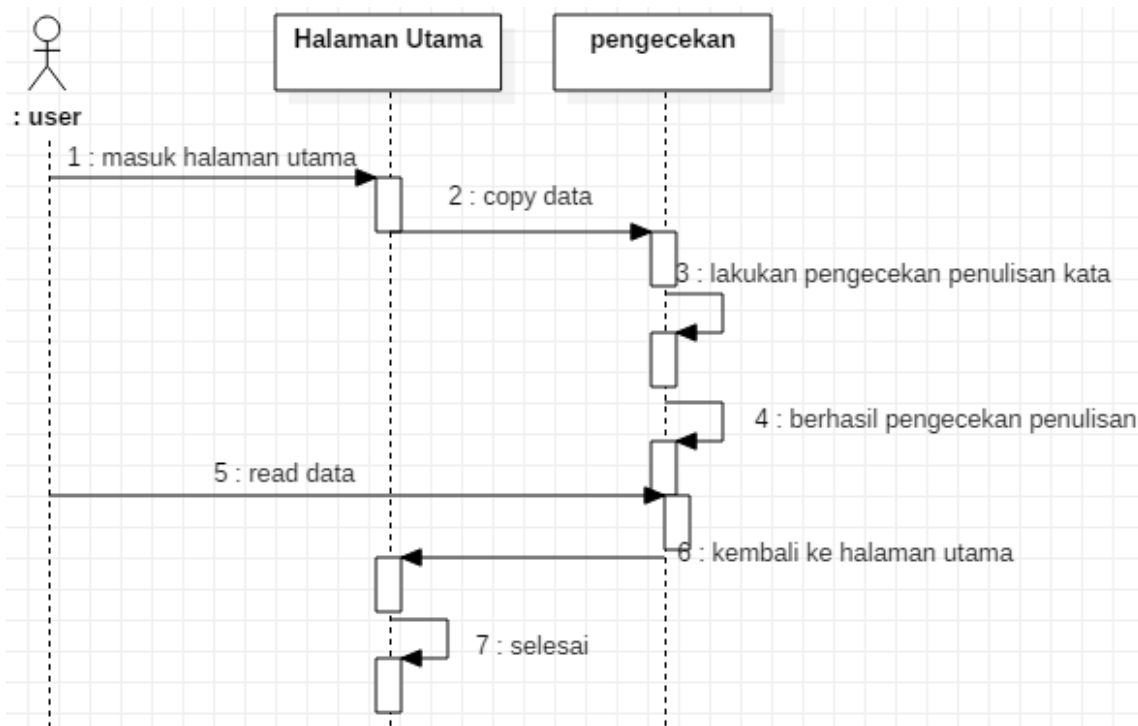
Gambar 3.10 Sequence Diagram Login User

Keterangan:

1. *User* masuk ke *form login*.
2. *System* meminta *UserName* dan password
3. *User* memasukkan *UserName* dan password / bikin akun terlebih dahulu.
4. Mengirim *UserName* dan password.
5. Melakukan Validasi data dengan *database*.
6. *Login* berhasil (Masuk ke halaman utama).

3.2.3.2 *Sequence diagram* Pengecekan Penulisan Data Ilmiah

Berikut ini merupakan *Sequence diagram* pengecekan yang menjelaskan proses pengecekan spelling.



Gambar 3.11 *Sequence Diagram* Pengecekan Spelling

Keterangan:

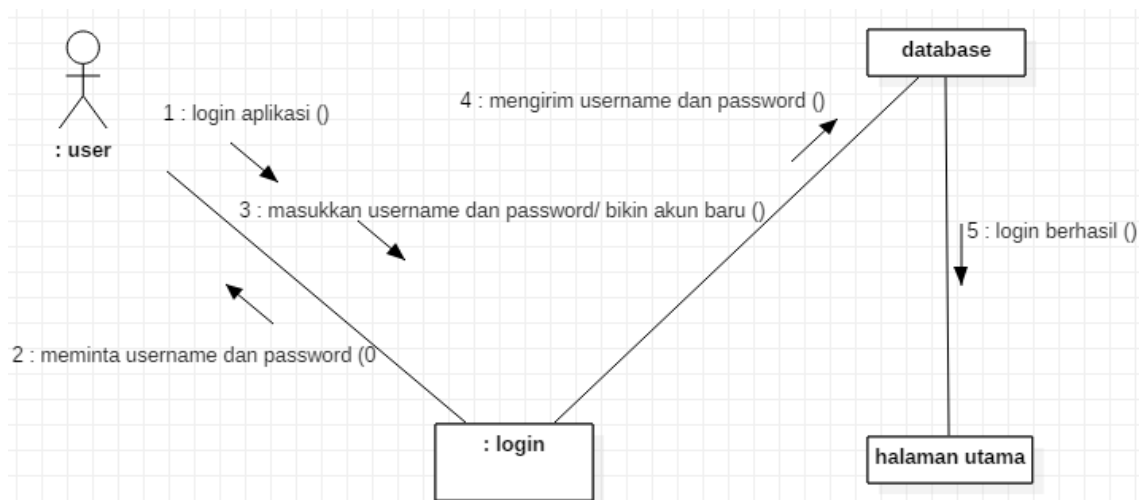
1. User akan masuk ke halaman utama.
2. User akan *Copy* data ilmiahnya.
3. Melakukan proses pengecekan *Spelling*
4. Data sudah berhasil di cek penulisan katanya.
5. User dapat *read* hasil pengecekan *spelling*
6. Selesai.

3.2.4 Collaboration Diagram

Collaboration diagram yaitu diagram yang mengelompokkan pesan pada kumpulan diagram sequent menjadi sebuah diagram. Dalam diagram tersebut terdapat *method* yang dijalankan antara objek yang satu dan objek lainnya. Di diagram kolaborasi ini, objek harus melakukan sinkronisasi pesan dengan serangkaian pesan-pesan lainnya.

3.2.4.1 Collaboration Diagram Login User

Berikut ini merupakan *Collaboration diagram* yang menjelaskan proses *login User*.



Gambar 3.12 Collaboration Diagram Login User

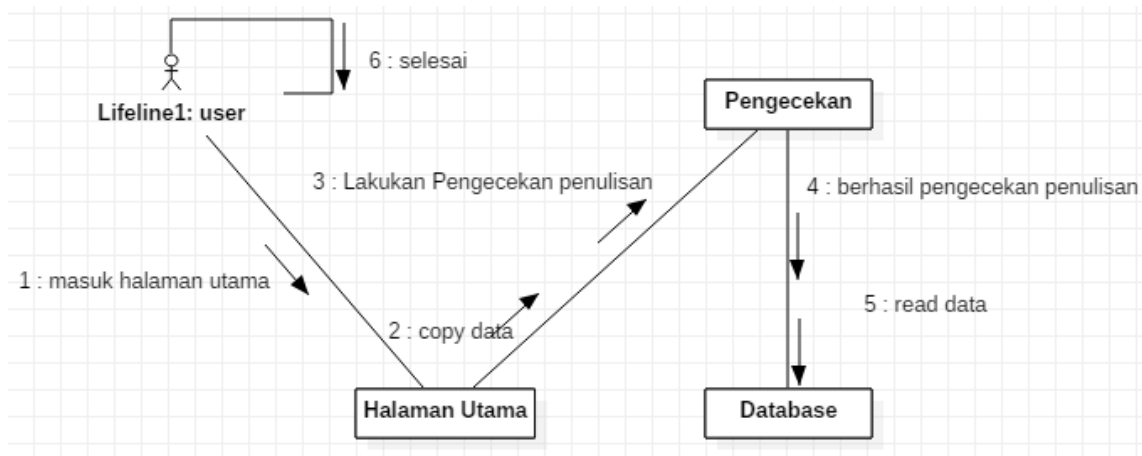
Keterangan :

1. User masuk ke form login.
2. System meminta UserName dan password
3. User memasukkan UserName dan password / bikin akun terlebih dahulu.

4. Mengirim *Username* dan password.
5. Melakukan Validasi data dengan *database*.
6. *Login* berhasil (Masuk ke halaman utama).

3.2.4.2 Collaboration Diagram Pengecekan Spelling

Berikut ini merupakan *Collaboration diagram* yang menjelaskan proses pengecekan *Spelling*



Gambar 3.13 Collaboration Diagram Pengecekan Spelling

Keterangan:

1. *User* akan masuk ke halaman utama.
2. *User* akan *Copy* data ilmiahnya.

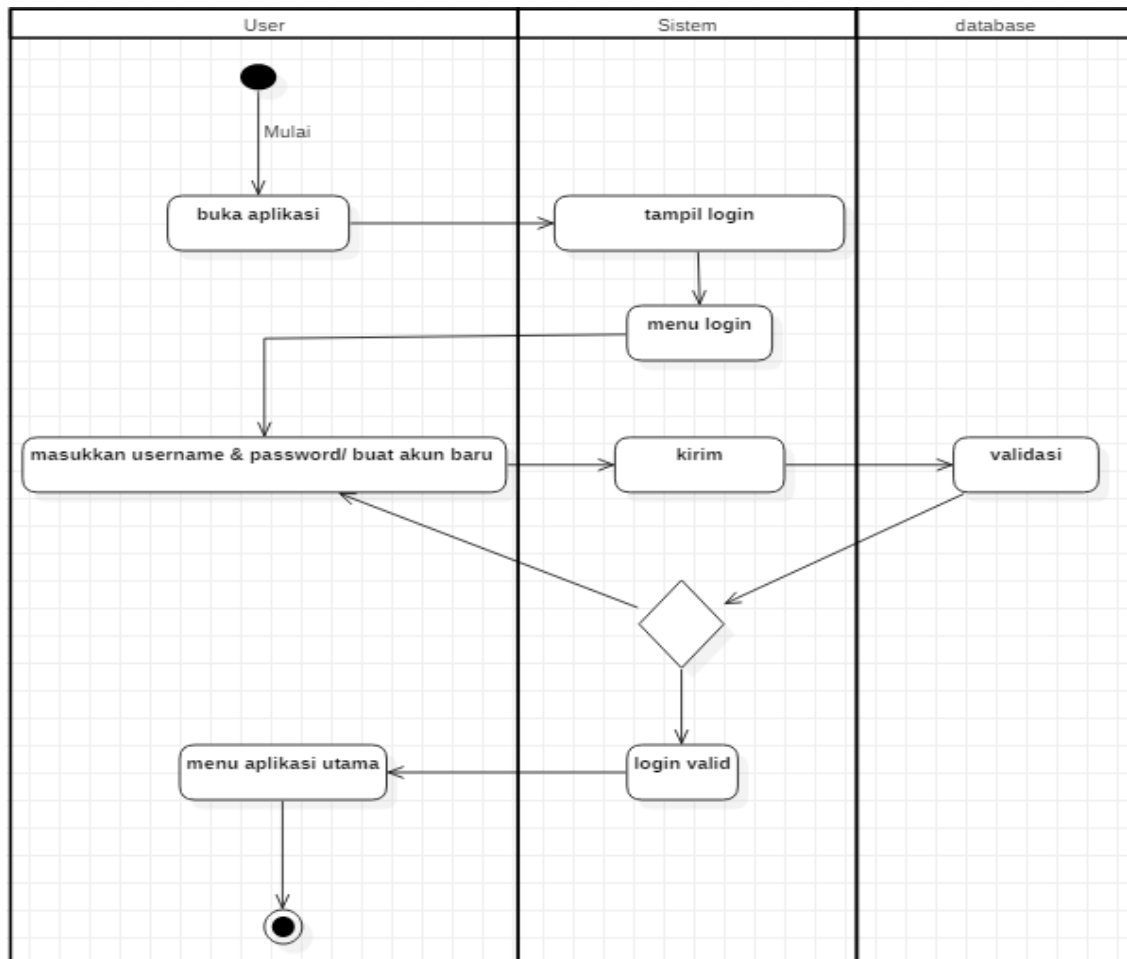
3. Melakukan proses pengecekan *spelling*
4. Data sudah berhasil di cek penulisan katanya.
5. *User* dapat *read* data hasil pengecekan *spealling*.

3.2.5 Activity Diagram

Activity diagram merupakan diagram yang berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

3.2.5.1 Activity Diagram Pada Login

Berikut ini merupakan *Activity diagram* pada *login* yang menjelaskan proses *Login User*.



Gambar 3.14 Activity Diagram Login

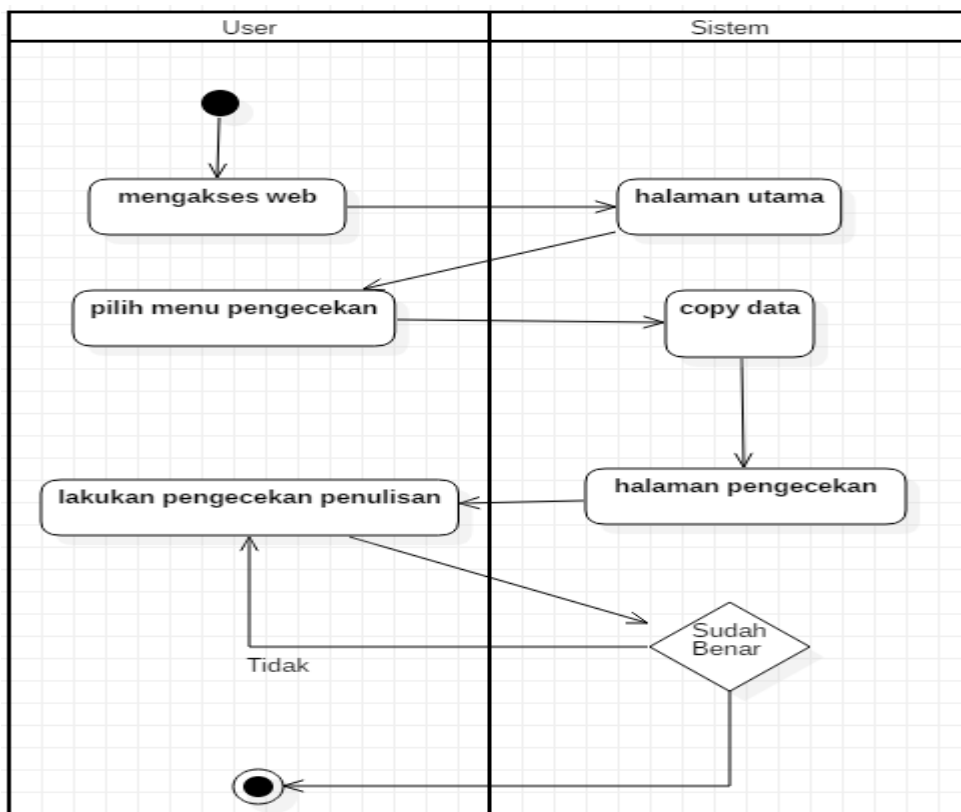
Keterangan Gambar 3.16 Activity Diagram Login:

1. *User* membuka aplikasi, masuk ke halaman *login*.
2. *User* memasukkan *UserName* dan password / buat akun baru.

3. Data dikirim ke sistem.
4. Data akan di validasi pada *database*.
5. Apabila berhasil maka *login* valid, apabila tidak maka *User* akan kembali memasukkan *UserName* dan password/ akun baru.
6. Menu halaman utama, proses selesai.

3.2.5.2 Activity Diagram Pada Pengecekan *Spealling*

Berikut ini merupakan *Activity diagram* pada Aplikasi Pengecekan *Spealling* yang menjelaskan proses pengecekan *Speallig* .



Gambar 3.15 Activity Diagram Pengecekan *Spealling*

Keterangan Gambar 3.17 Activity Diagram Pengecekan *Spelling* :

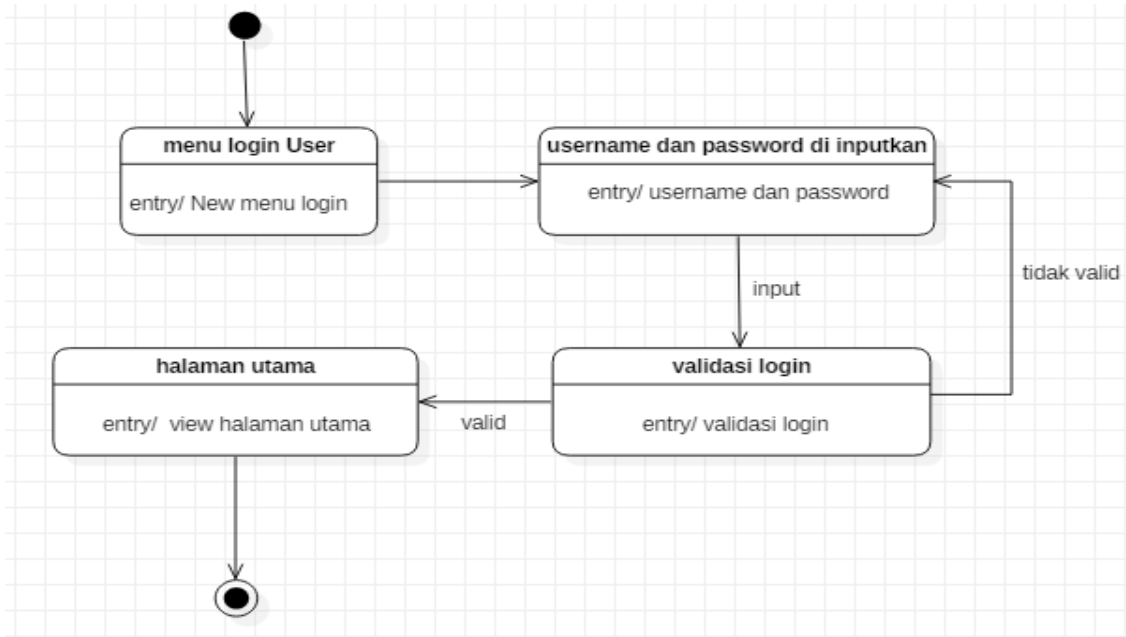
1. *User* mengakses web pengecekan *Spealling* , lalu akan masuk ke halaman utama.
2. *User* memilih menu pengecekan, yaitu menu pengecekan *Spealling*.
3. *User* akan *Copy* data yang akan dicek penulisannya. Maka akan masuk ke halaman pengecekan.
4. Melakukan proses pengecekan *Spealling*.
5. Apabila data sudah benar, maka proses tersebut telah selesai, apabila belum maka proses pengecekan penulisan lagi.
6. *User* dapat read data tersebut, dan selesai.

3.2.6 State chart Diagram

Menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulan yang diterima, *State chart* diagram digunakan untuk memodelkan *behavior/methode (lifecycle)* sebuah kelas atau objek, Memperlihatkan urutan kejadian sesaat (*state*) yang dilalui sebuah objek, transisi dari sebuah *state* ke *state* lainnya.

3.2.6.1 State chart Diagram Pada Login

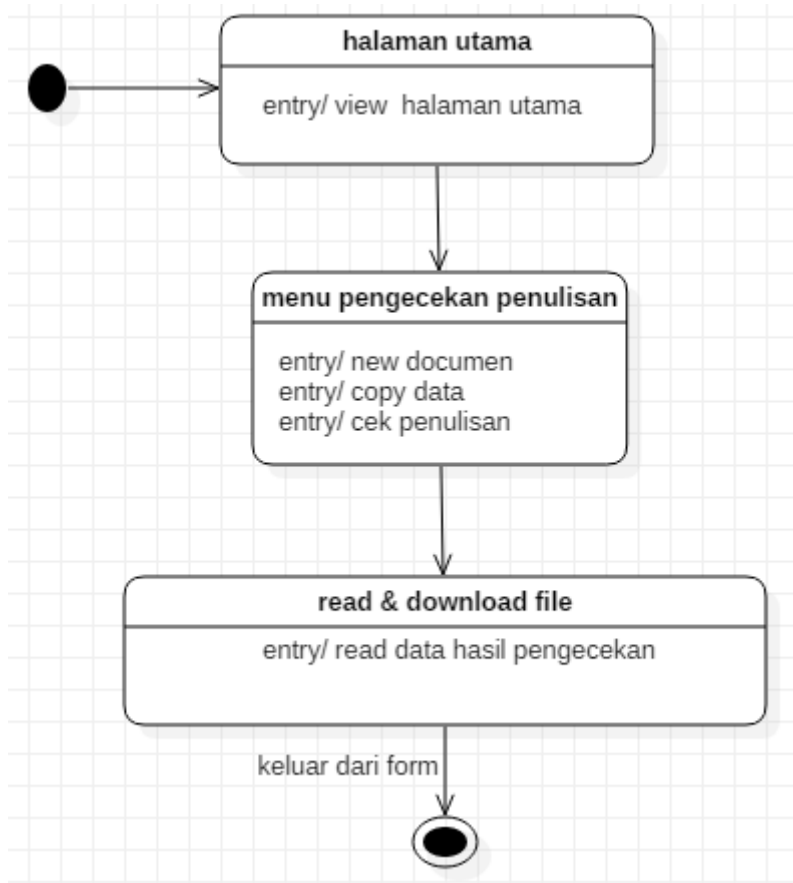
Berikut ini merupakan *State chart diagram* pada *login* yang menjelaskan proses *login User*.



Gambar 3.16 State chart Diagram Login

3.2.6.2 State chart Diagram Pada Pengecekan *Spealling*

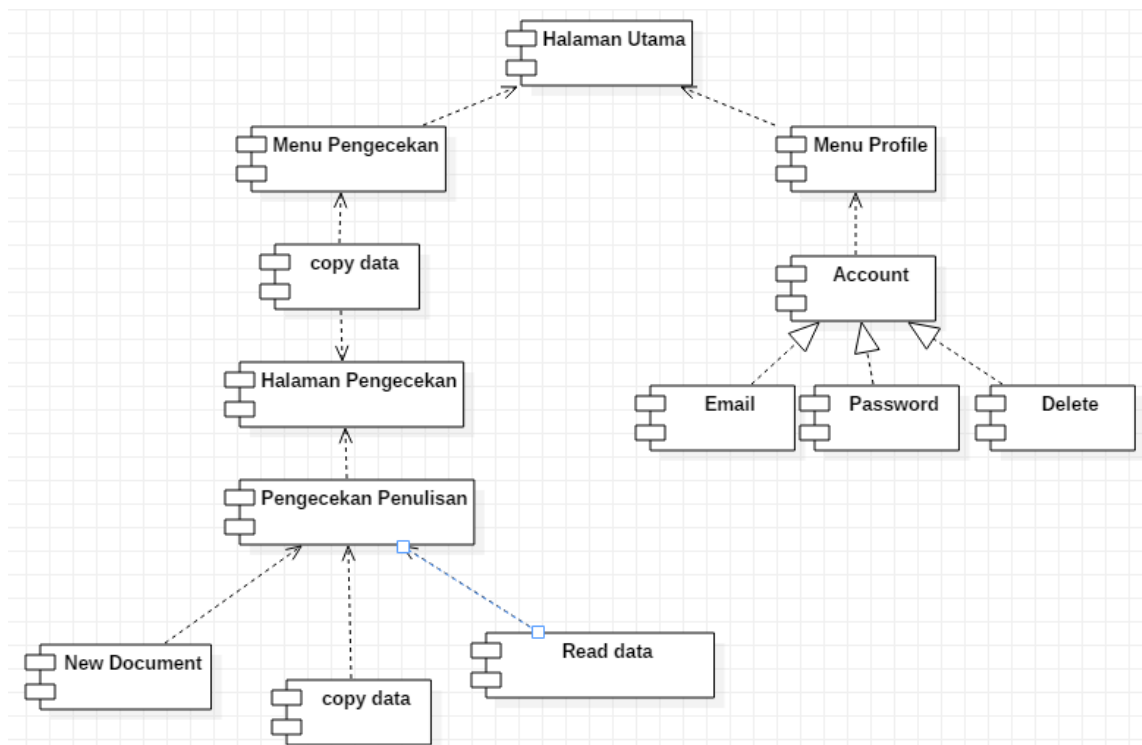
Berikut ini merupakan *State chart diagram* pada pengecekan penulisan yang menjelaskan pengecekan penulisan pada data ilmiah.



Gambar 3.17 State chart Diagram Pengecekan Spealling

3.2.7 Component Diagram

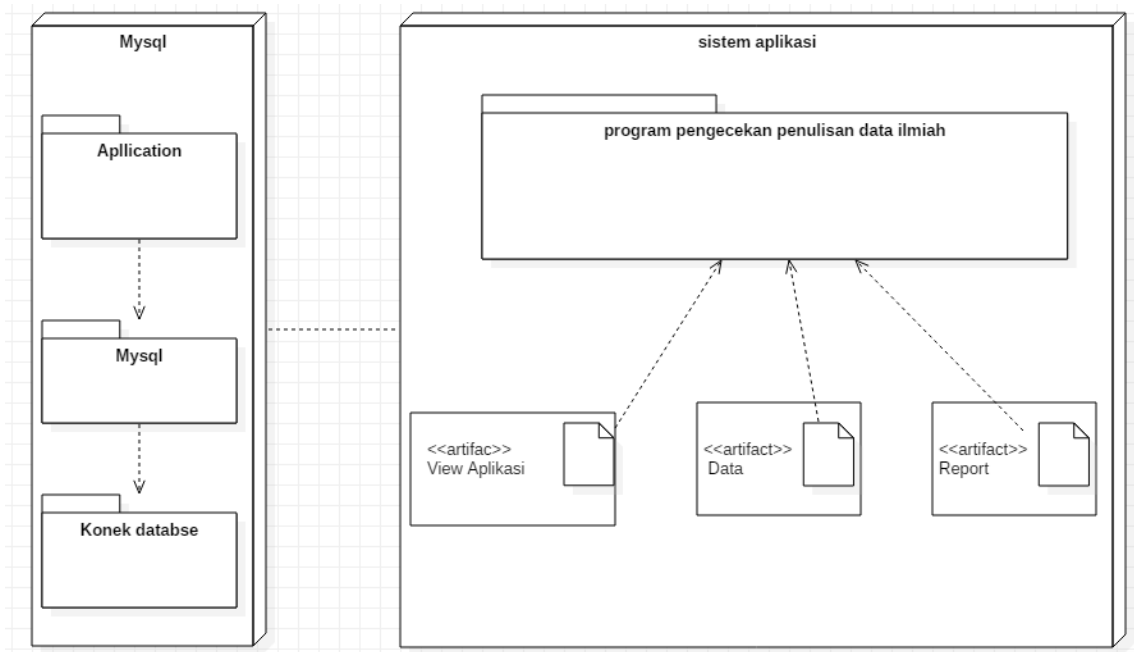
Component diagram yang terdapat dalam aplikasi pengecekan *Spealling* dalam aplikasi terdapat menu pengecekan dan menu *prodata*. Aplikasi tersebut ter koneksi dengan *database connection*.



Gambar 3.18 Component Diagram Aplikasi Pengecekan

3.2.8 Deployment Diagram

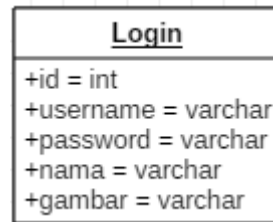
Deployment Diagram yang terdapat dalam aplikasi pengecekan penulisan ini menyebarkan setiap komponennya dengan menggunakan koneksi dari aplikasi lalu dengan menyambungkan dengan *MySQL* dihubungkan melalui *database* koneksi. Di dalam program aplikasi rekam medis itu pun terdapat *view* aplikasi, data dan *report* laporan yang akan muncul.



Gambar 3.19 Deployment Diagram Aplikasi Pengecekan

3.2.9 Object Diagram

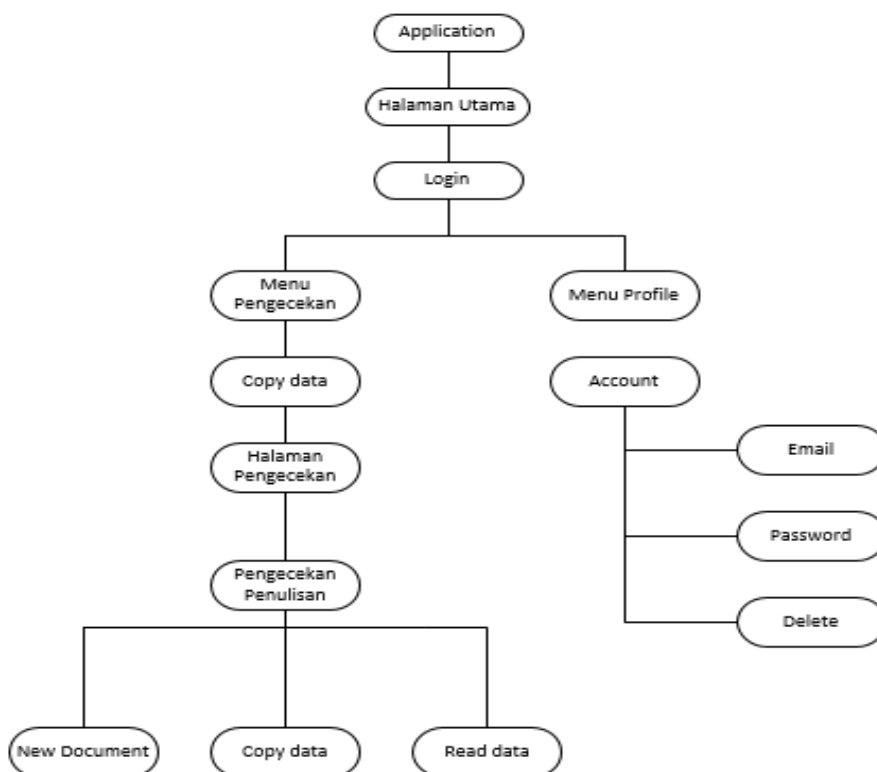
Objek diagram yang terdapat dalam aplikasi pengecekan penulisan hanya terdapat objek pada proses log in saja.



Gambar 3.20 Object Diagram Aplikasi Pengecekan

3.2.10 Struktur Diagram

Berikut merupakan struktur diagram yang menjelaskan alur struktur dari aplikasi pengecekan *Spealling*.



Gambar 3.21 Struktur Diagram Aplikasi Pengecekan

3.3 Perancangan Antarmuka

3.3.1 User Interface Login Pengguna

<div>Login With Google</div>	
Email:	<input type="text"/>
Password:	<input type="password"/>
	<div>Masuk</div>

Gambar 3.22 User Interface Login Pengguna

Keterangan Gambar :

Command Button 1 : *Tombol login dengan akun google*

Label 1 : *Email*

Label 2 : *password*

Textbox 1 : *Input Email*

Textbox 2 : *Input Password*

Command Button 1 : *Tombol Masuk*

3.3.2 User Interface Halaman Iklan

Judul	Login
Animasi	

Gambar 3.23 User Interface Halaman Iklan

Keterangan Gambar :

Label 1 : *Judul*

Button1 : *Login*

Textbox 1 : *Animasi bergerak*

3.3.3 User Interface Halaman Pengecekan

Judul	
Pengecekan	<div></div> <div>Check here to spell check</div>

Gambar 3.24 User Interface Halaman Pengecekan

Keterangan Gambar :

- Label 1* : *Judul*
- Label 2* : *Menu pengecekan*
- Textbox 1* : *Form Input Pengecekan*
- Button 1* : *Check here to spell check*

3.3.4 User Interface Halaman Profile

Judul	Profile
Profile	Account
	Data Account google

Gambar 3.25 User Interface Halaman Profile

Keterangan Gambar :

- Label 1* : *Judul*
- Label 2* : *Menu Profile*
- Label 3* : *Halaman Profile*
- Label 4* : *Account*
- Label 5* : *Data Google Account*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari Pengembangan Aplikasi *SPelling checker* ialah adalah aplikasi pengecekan penulisan ini dapat mengurangi penulisan kesalahan kata pada saat pembuatan sebuah karya ilmiah, seperti makalah, skripsi, tesis maupun yang lainnya karena sistem menyediakan fitur cek tulisan kata-kata apakah ada kesalahan penulisan pada kata yang dimasukkan, yang akan digunakan dengan kondisi yang disediakan.

5.2 Saran

Saran yang dapat diberikan berdasarkan hasil yang telah dicapai maupun untuk pengembangan aplikasi pada masa yang akan datang, antara lain :

Appendix A

Form Penilaian Jurnal

gambar A.1 dan A.2 merupakan contoh bagaimana reviewer menilai jurnal kita.

Gambar A.1: Form nilai bagian 1.

Gambar A.2: form nilai bagian 2.

Appendix B

FAQ

M : Kalo Intership II atau TA harus buat aplikasi ? D : Ga harus buat aplikasi tapi harus ngoding

M : Pa saya bingung mau ngapain, saya juga bingung mau presentasi apa? D : Makanya baca de, buka jurnal topik ‘ganteng’ nah kamu baca dulu sehari 5 kali ya, 4 hari udah 20 tuh. Bingung itu tanda kurang wawasan alias kurang baca.

M : Pa saya sudah cari jurnal terindeks scopus tapi ga nemu. D : Kamu punya mata de? coba dicolok dulu. Kamu udah lakuin apa aja? tolong di list laporkan ke grup Tingkat Akhir. Tinggal buka google scholar klik dari tahun 2014, cek nama jurnalnya di scimagojr.com beres.

M : Pa saya belum dapat tempat intership, jadi ga tau mau presentasi apa? D : kamu kok ga nyambung, yang dipresentasikan itu yang kamu baca bukan yang akan kamu lakukan.

M : Pa ini jurnal harus yang terindex scopus ga bisa yang lain ? D : Index scopus menandakan artikel tersebut dalam standar semantik yang mudah dipahami dan dibaca serta bukan artikel asal jadi. Jika diluar scopus biasanya lebih sukar untuk dibaca dan dipahami karena tidak adanya proses review yang baik dan benar terhadap artikel.

M : Pa saya tidak mengerti D : Coba lihat standar alasan M : Pa

saya bingung D : Coba lihat standar alasan

M : Pa saya sibuk D : Mbahmu....

M : Pa saya ganteng D : Ndasmu....

M : Pa saya kece D : wes karepmu lah....

Biasanya anda memiliki alasan tertentu jika menghadapi kendala saat proses bimbingan, disini saya akan melakukan standar alasan agar persepsi yang diterima sama dan tidak salah kaprah. Penggunaan kata alasan tersebut antara lain :

1. Tidak Mengerti : anda boleh menggunakan alasan ini jika anda sudah melakukan tahapan membaca dan meresumekan 15 jurnal. Sudah mencoba dan mempraktekkan teorinya dengan mencari di youtube dan google minimal 6 jam sehari selama 3 hari berturut-turut.

2. Bingung : anda boleh mengatakan alasan bingung setelah maksimal dalam berusaha menyelesaikan tugas bimbingan dari dosen(sudah dilakukan semua). Anda

belum bisa mengatakan alasan bingung jika anda masih belum menyelesaikan tugas bimbingan dan poin nomor 1 diatas. Setelah anda menyelesaikan tugas bimbingan secara maksimal dan tahap 1 poin diatas, tapi anda masih tetap bingung maka anda boleh memakai alasan ini.

DAFTAR PUSTAKA