**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : ARTHA PRADYTA**

**NRP : 5110100109**

**DOSEN WALI : Anny Yuniarti, S.Kom., M.Comp.Sc.**

**DOSEN PEMBIMBING : 1. Dr. Ir. Siti Rochimah, M.T.  
 2. Umi Laili Yuhana, S.Kom., M.Sc.**

# JUDUL TUGAS AKHIR

“Rancang Bangun Aplikasi Sistem Validasi Data Berkas Beasiswa”

# LATAR BELAKANG

Pemberian beasiswa merupakan program kerja tahunan yang ada di setiap universitas maupun perguruan tinggi. Program beasiswa ini ditawarkan kepada mahasiswa yang berprestasi dan kepada mahasiswa yang kurang mampu seperti yang tertuang dalam Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional [1]. Di dalam Undang-Undang tersebut bagian Bab V Pasal 12 (1.c), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan beasiswa bagi yang berprestasi yang orang tuanya kurang mampu membiayai pendidikannya serta pada Pasal 12 (1.d), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan biaya pendidikan bagi mereka yang orang tuanya kurang mampu membiayai pendidikannya. Selain itu, dalam UUD 1945 pasal 31 ayat 1 [2] juga menerangkan bahwa tiap-tiap warga Negara berhak mendapatkan pengajaran.

Berdasarkan pasal-pasal tersebut, bisa disimpulkan bahwa pemerintah pusat maupun pemerintah daerah wajib memberikan layanan dan kemudahan, serta menjamin terselenggaranya pendidikan yang bermutu bagi setiap warga Negara tanpa adanya diskriminasi ras, kekayaan, serta kedudukan sosial. Akan tetapi, untuk menyelenggarakan pendidikan yang bermutu diperlukan juga biaya pendidikan yang cukup besar di mana tidak setiap warga Negara mampu membiayai pendidikannya. Oleh karena itu, bagi setiap peserta didik dalam setiap satuan pendidikan berhak mendapatkan bantuan biaya pendidikan bagi yang orang tuanya tidak mampu membiayai pendidikannya, dan berhak mendapatkan beasiswa bagi mereka yang berprestasi.

Berdasarkan peraturan Direktorat Jenderal Pendidikan Tinggi (DIKTI), semua pihak penyelenggara pendidikan tinggi tidak terkecuali Institut Teknologi Sepuluh Nopember Surabaya (ITS) diwajibkan untuk menyelenggarakan program pemberian beasiswa, baik beasiswa yang diberikan pemerintah ataupun pihak swasta di mana program pemberian beasiswa ini harus dilaksanakan sesuai dengan prinsip 3T yaitu Tepat sasaran, Tepat jumlah, dan Tepat waktu.

Di Institut Teknologi Sepuluh Nopember Surabaya (ITS) sendiri sudah ada sistem yang menangani program pemberian beasiswa. Namun untuk menyelenggarakan program pemberian beasiswa sesuai prinsip 3T terdapat kendala pada sistem lama di antaranya lamanya proses antara pendaftaran calon penerima beasiswa sampai dengan proses seleksi calon penerima beasiswa. Hal ini sangat sering terjadi ketika terdapat banyak sekali program beasiswa yang memberikan pengumuman secara mendadak dan terkadang banyak mahasiswa yang baru mengetahui info beasiswa di saat mendekati batas akhir pengumpulan berkas sehingga sering terjadi mahasiswa kehilangan peluang untuk mengikuti seleksi beasiswa karena mepetnya waktu untuk menyiapkan syarat-syarat yang diperlukan. Selain itu sering sekali ketika seorang mahasiswa ingin mencoba ikut beberapa seleksi beasiswa sekaligus, terdapat beberapa berkas yang sama yang harus diunggah berkali-kali karena program-program beasiswa yang mahasiswa ikuti memiliki beberapa persyaratan yang sama sehingga masalah seperti ini sebenarnya bisa diakali dengan dibuatkannya sebuah *database* mahasiswa yang menyimpan data-data maupun berkas mahasiswa secara lengkap di mana kedepannya mahasiswa tidak perlu repot-repot lagi ketika membutuhkan data tersebut untuk mendaftar beasiswa.

Adapun permasalahan lain yang terdapat dalam sistem lama yaitu pada proses pengunggahan berkas-berkas mahasiswa berbentuk *file* gambar untuk memenuhi persyaratan pendaftaran beasiswa seperti transkrip, kartu keluarga, surat keterangan miskin, surat keterangan gaji orang tua, dan lain sebagainya. Ketika berkas tersebut diunggah oleh mahasiswa, sistem tidak bisa melakukan validasi data berkas secara otomatis sehingga diperlukan pengecekan secara manual oleh manusia dengan melihat secara langsung *file* yang diunggah oleh mahasiswa. Hal ini tentu saja tidak ada bedanya dengan proses pengumpulan berkas secara konvensional karena masih melibatkan manusia untuk melakukan validasi data berkas padahal seharusnya validasi data berkas tersebut bisa dilakukan oleh sistem.

Oleh karena itu diperlukan rancang bangun ulang terhadap sistem pemberian beasiswa yang dimiliki ITS agar dapat digunakan secara nyaman, aman dan lebih tinggi tingkat kebergunaannya. Selain itu juga dapat meringankan beban kerja pihak penyeleksi beasiswa, mempersingkat waktu penyeleksian, serta dapat meningkatkan kualitas keputusan dalam penentuan penerima beasiswa.

# RUMUSAN MASALAH

1. Cara memvalidasi data berkas persyaratan beasiswa yang diunggah oleh mahasiswa.
2. Menerapkan metode *Optical Character Recognition* pada sistem validasi data berkas persyaratan beasiswa.
3. Menerapkan algoritma *Winnowing, Levenshtein Distance*, *Longest Common Subsequence*, dan *Soundex* pada sistem validasi data berkas persyaratan beasiswa.

# BATASAN MASALAH

Pada dasarnya permasalahan dalam sistem beasiswa cukup luas, tetapi agar sesuai dengan yang telah direncanakan sebelumnya, diperlukan batasan-batasan agar tujuan dalam Tugas Akhir ini dapat tercapai. Adapun batasan-batasannya adalah sebagai berikut.

1. Studi kasus yang dibahas adalah SI Beasiswa di Perguruan Tinggi X.
2. Data yang digunakan pada fase awal adalah data sekunder (*dummy*).
3. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman ASP.NET dan *database* menggunakan MySQL.
4. Tingkat keberhasilan validasi tergantung dengan kejelasan gambar yang diunggah.
5. Algoritma pencocokan *string* yang digunakan adalah algoritma *Winnowing*, *Levenshtein Distance*, *Longest Common Subsequence*, dan *Soundex*.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dalam pembuatan Tugas Akhir ini adalah sebagai berikut.

1. Meningkatkan efisiensi kinerja pihak penyeleksi beasiswa dalam penyeleksian calon penerima beasiswa.
2. Memudahkan kinerja pihak penyeleksi beasiswa dalam memvalidasi data berkas yang diunggah mahasiswa.
3. Menerapkan metode *Optical Character Recognition* dan algoritma *Winnowing, Levenshtein Distance*, *Longest Common Subsequence*, serta *Soundex* untuk validasi sebuah berkas.

# MANFAAT TUGAS AKHIR

Manfaat yang didapat dalam pembuatan Tugas Akhir ini adalah sebagai berikut.

1. Dapat membantu kinerja tim penyeleksi beasiswa dalam melakukan penyeleksian dan validasi data berkas beasiswa.
2. Dapat mempercepat proses penyeleksian beasiswa.
3. Dapat mempermudah mahasiswa untuk mengikuti proses seleksi beasiswa.

# TINJAUAN PUSTAKA

## Beasiswa

Pengertian beasiswa menurut Kamus Besar Bahasa Indonesia yaitu tunjangan uang yang diberikan kepada pelajar atau mahasiswa sebagai bantuan biaya belajar. Sedangkan menurut istilah lain, beasiswa merupakan tunjangan uang, diberikan kepada pelajar-pelajar baik dengan cuma-cuma atau sebagai persekot tidak berbunga, untuk menyelesaikan pendidikannya [3].

Pada dasarnya, beasiswa adalah penghasilan bagi yang menerimanya. Beasiswa ini telah sesuai dengan ketentuan pasal 4 ayat (1) UU PPh/2000 di mana dalam pasal tersebut disebutkan bahwa beasiswa merupakan tambahan kemampuan ekonomis dengan nama dan bentuk apa pun yang diterima atau diperoleh dari sumber Indonesia atau luar Indonesia yang dapat digunakan untuk konsumsi pribadi atau menambah kekayaan Wajib Pajak (WP). Karena beasiswa bisa diartikan sebagai menambah kemampuan ekonomis bagi penerimanya, berarti beasiswa juga bisa diartikan sebagai penghasilan [4].

## Algoritma *Winnowing*

Algortima *Winnowing* merupakan algoritma untuk membandingkan dokumen *fingerprinting* dengan menggunakan teknik *hashing* [5]. Masukan dari algoritma ini adalah dokumen teks yang diproses sehingga menghasilkan keluaran berupa kumpulan nilai-nilai *hash*. Nilai *hash* merupakan nilai numerik yang terbentuk dari perhitungan nilai ASCII dari setiap karakter. Kumpulan-kumpulan nilai *hash* tersebut selanjutnya disebut *fingerprint*. *Fingerprint* inilah yang digunakan untuk membandingkan kemiripan antar dokumen teks. Algoritma pencocokan *string* ini nantinya digunakan untuk validasi dengan membandingkan data yang dimasukkan mahasiswa ke sistem dengan data yang didapat dari berkas persyaratan mahasiswa dalam bentuk gambar yang sudah diekstraksi.

Berikut ini merupakan langkah-langkah dasar dalam algoritma *Winnowing*.

* Penghapusan karakter-karakter yang tidak relevan (*whitespace insensitivity*), antara lain spasi atau tanda baca.
* Pembentukan rangkaian *gram* dengan ukuran k.
* Penghitungan nilai *hash* dari setiap *gram* dengan proses *Rolling Hash*.
* Membagi ke dalam *window* tertentu.
* Pemilihan beberapa nilai *hash* menjadi *document fingerprinting*.
* Menentukan persentase kesamaan antara 2 dokumen dengan persamaan *Jaccard Coefficient*.

*Rolling Hash* sendiri merupakan sebuah fungsi yang digunakan untuk menghasilkan nilai dari rangkaian *gram* di mana fungi *hash* didefinisikan seperti Persamaan 1.

Di mana :

c : nilai ASCII karakter,

b : basis bilangan prima,

k : nilai *k-gram* / banyaknya karakter pada *gram*.

Untuk mendapatkan nilai *hash gram* berikutnya dapat dilakukan dengan cara seperti Persamaan 2.

Pada perhitungan *hash* dari *gram* ke-n, nilai *hash gram* n-1 dikurangi dengan nilai karakter pertama dari *gram* n-1 kemudian ditambahkan dengan nilai karakter terakhir dari *gram* ke-n. Dengan begitu tidak perlu melakukan iterasi dari indeks pertama sampai dengan terakhir untuk menghitung nilai *hash* untuk *gram* ke-2 sampai terakhir. Sedangkan untuk menghitung persentase kemiripan (*similarity*), bisa menggunakan rumus *Jaccard’s Similarity Coefficient* seperti Persamaan 3.

Di mana D(A,B) adalah nilai *similarity*, adalah jumlah dari *fingerprints* yang sama dari dokumen 1 dan 2, dan adalah jumlah *fingerprints* dari dokumen 1 dan dokumen 2. Sebagai contoh a={1,2,4}, b={1,2,4,7,8} maka, = {1,2,4} dan = {1,2,4,7,8} sehingga D(a,b) = 3/5 \* 100 = 60.

## Algoritma *Soundex* dan Normalisasi *String*

Algoritma *Soundex* pertama kali dipatenkan oleh Margaret O’Dell dan Robert C. Russell pada tahun 1918 [6]. Algoritma *Soundex* mengambil masukan berupa sebuah *string* yang mengidentifikasi apakah sepasang kata tersebut mirip secara fonetik. *String* ini disebut dengan kode fonetis. Fonetik sendiri merupakan ilmu yang menyelidiki bunyi bahasa tanpa melihat fungsi bunyi itu sebagai pembeda makna dalam suatu bahasa. Pencocokan *string* fonetik adalah suatu teknik pencocokan *string* yang membandingkan suatu *string* dengan *string* yang lain berdasarkan kode fonetis masing-masing [7]. Sebuah *string* yang berbeda namun mempunyai cara pengucapan yang sama, akan memiliki kode fonetis yang sama. Contohnya adalah “Djono” dan “Jono”.

Metode yang digunakan algoritma *Soundex* berdasarkan pada klasifikasi fonetis dari suara cara berbicara manusia (*human speech sound*) yaitu *labial*, *dental*, *alveolar*, *palato-alveolar*, *palatal*, *velar*, dan *glottal* [6]. Berikut ini merupakan langkah-langkah algoritma *Soundex* dalam menghasilkan kode fonetis dari sebuah *string* masukan.

* Mengubah semua huruf menjadi huruf kapital, dan hilangkan tanda baca.
* Mempertahankan huruf pertama pada kata tersebut.
* Mengubah huruf lainnya menjadi kode fonetis berdasarkan aturan yang berlaku.
* Menghapus semua pasangan dari kode fonetis yang berurutan.
* Menghapus semua kode fonetis 0.
* Menulis empat posisi pertama yang mengikuti pola: <*uppercase letter*><*digit*><*digit*><*digit*>. Ini adalah kode fonetis sebagai keluaran. Jika kode fonetis tidak sampai empat karakter, maka ditambahkan *digit* 0 sampai menjadi empat karakter.

## Algoritma *Levenshtein Distance*

Algoritma *Levenshtein Distance* dinamakan berdasarkan penemunya, yaitu Dr. Vladimir Levenshtein. Dr. Vladimir Levenshtein dikenal sebagai bapak dari teori koding (*father of coding theory*) di Rusia [8]. *Levenshtein Distance* saat ini adalah dasar dari banyak aplikasi komputer di bidang pemeriksaan ejaan.

*Levenshtein Distance* adalah jumlah minimal operasi yang dibutuhkan untuk mengubah suatu *string* ke *string* yang lain. Dengan algoritma ini, dapat diketahui tingkat perbedaan dua buah *string* dalam representasi angka. Berikut ini operasi yang dilakukan dalam algoritma *Levenshtein Distance* [9].

* *Insertion*

Merupakan proses penyisipan sebuah karakter ke dalam sebuah *string* tertentu. Contohnya, penyisipan karakter ‘k’ ke dalam *string* “epala” pada posisi sebelum karakter ‘e’ akan mengubah *string* “epala” emnjadi *string* “kepala”.

* *Deletion*

Merupakan proses penghapusan sebuah karakter dari sebuah *string* tertentu. Contohnya, penghapusan karakter ‘a’ pada posisi terakhir *string* “kepala” akan mengubah *string* “kepala” menjadi *string* “kepal”.

* *Substitution*

Merupakan proses penggantian sebuah karakter dari sebuah *string* dengan karakter lain. Contohnya, penggantian karakter ‘e’ dengan karakter ‘a’ pada *string* “kepale” akan mengubah *string* “kepale” menjadi *string* “kepala”.

Algoritma *Levenshtein Distance* sendiri menggunakan konsep *dynamic programming* dalam penerapannya. Algoritma ini menggunakan sebuah matriks berukuran panjang dari *string-string* yang menjadi masukan. *Levenshtein Distance* yang dihasilkan adalah angka yang terletak di kotak paling bawah kanan dari matriks, sebagai jumlah minimum operasi yang dibutuhkan untuk mengubah *string* awal menjadi *string* target.

## Algoritma *Longest Common Subsequence*

*Subsequence* adalah sebuah rangkaian yang dapat diperoleh dari rangkaian lain dengan cara menghapus beberapa elemen tertentu tanpa mengubah urutan dari elemen lain yang tersisa [10]. Misalkan ada rangkaian X dan Y, rangkaian S dikatakan sebuah *common subsequence* dari X dan Y apabila S adalah *subsequence* dari keduanya. *Longest Common Subsequence* (LCS) merupakan *common subsequence* terpanjang dari seluruh *common subsequence*. Contohnya adalah sebagai berikut.

X = {M,A,K,A,N}.

Y = {M,A,L,A,M}.

Contoh *common subsequence* dari X dan Y adalah S = {M,A,A} dan T = {M,A}. Karena S merupakan *subsequence* yang lebih panjang, maka *longest common subsequence* dari X dan Y adalah S. Algoritma untuk masalah LCS ini berdasarkan teorema berikut [11].

Misal X = <x1,x2,…,xm> dan Y = <y1,y2,…,yn> adalah rangkaian dan Z = <z1,z2,…,zk> adalah suatu LCS dari X dan Y.

1. Jika xm = yn, maka zk = xm = yn dan Zk-1 adalah suatu LCS dari Xm-1 dan Yn-1.
2. Jika xm ≠ yn, maka zk ≠ xm mengimplikasikan bahwa Z adalah suatu LCS dari Xm-1 dan Y.
3. Jika xm ≠ ym, maka zk ≠ yn mengimplikasikan bahwa Z adalah suatu LCS dari X dan Yn-1.

## *Optical Character Recognition*

OCR (*Optical Character Recognition*) merupakan sebuah teknologi konversi dari gambar ke bentuk teks. Teknologi ini biasanya digunakan untuk mengambil data dari berkas-berkas berbentuk gambar seperti dokumen paspor, faktur, laporan bank, kartu nama, dan sebagainya. Dalam pemrosesan OCR, gambar akan dianalisis pada daerah terang dan gelap dalam rangka mengidentifikasi setiap huruf abjad maupun angka numerik [12]. Ketika karakter sudah diketahui, maka tiap kata tersebut akan dirubah ke dalam bentuk ASCII. Dalam hal ini mesin komputer akan merubah kode ASCII tersebut menjadi berupa karakter maupun angka.

Karena kemampuannya dalam hal ekstraksi data dari gambar inilah, maka teknologi ini akan digunakan untuk mengambil data-data dari berkas persyaratan mahasiswa yang telah diunggah ke sistem seperti laporan slip gaji, kartu keluarga, dan sebagainya.

Berikut ini merupakan langkah-langkah alur kerja sistem OCR secara umum.

* *Data Capture*

Merupakan proses konversi suatu dokumen menjadi suatu *file* gambar *bitmap*.

* *Preprocessing*

Merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada *input* untuk proses selanjutnya. Beberapa contoh *preprocessing* adalah *noise filtering*.

* *Segmentation*

Merupakan proses pemisahan area pengamatan pada tiap karakter yang akan dideteksi.

* *Normalization*

Merupakan proses pengubahan dimensi area pengamatan tiap karakter dan ketebalan karakter. Dalam OCR algoritma yang digunakan dalam proses ini adalah algoritma *scaling* dan *thinning*.

* *Feature Extraction*

Merupakan proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.

* *Recognition*

Merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri yang diperoleh dengan ciri-ciri karakter yang ada pada *database*.

* *Postprocessing*

Merupakan proses koreksi ejaan.

## ASP .NET

ASP.NET merupakan singkatan dari *Active Server Pages NET* yang merupakan seperangkat alat pengembangan *Website* yang ditawarkan oleh Microsoft. Program seperti Visual Studio .NET dan Visual Web Developer memungkinkan pengembang *Website* untuk membuat *Website* dinamis menggunakan antarmuka visual [13]. Sebagai anggota dari .NET, ASP.NET adalah *tool* yang sangat berguna bagi programmer yang bertujuan untuk memudahkan programmer dalam membuat *Website* yang dinamis dan disertai dengan bahasa VB dan C#.

* 1. ***Web Service***

Menurut W3C [14] *Web Service* adalah suatu sistem perangkat lunak yang didesain untuk mendukung interaksi mesin ke mesin pada suatu jaringan. Sistem lain yang berinteraksi dengan *Web Service* dilakukan melalui antar muka menggunakan pesan seperti pada SOAP. Pada umumnya pesan ini melalui HTTP dan XML yang merupakan salah satu standar *Website*. Perangkat Lunak aplikasi yang ditulis dalam berbagai bahasa pemrograman dan berjalan pada berbagai platform dapat menggunakan *Web Service* untuk pertukaran data pada jaringan komputer seperti Internet dalam cara yang serupa dengan komunikasi *inter-process* pada komputer tunggal. Interoperabilitas ini (sebagai contoh, antara Java dan Python, atau Microsoft Windows dan aplikasi Linux) adalah dalam kaitan dengan penggunaan dari *open standard.*

Dalam hal ini nantinya *Web Service* digunakan sebagai perantara pertukaran data berkas antara SI Beasiswa dengan aplikasi validasi berkas beasiswa.

# RINGKASAN ISI TUGAS AKHIR

Program pemberian beasiswa merupakan sebuah program rutin yang diwajibkan oleh Direktorat Jenderal Pendidikan Tinggi (DIKTI) untuk diselenggarakan semua pihak penyelenggara pendidikan tinggi tidak terkecuali Institut Teknologi Sepuluh Nopember Surabaya (ITS). Di ITS Surabaya sendiri sudah ada sistem yang menangani program pemberian beasiswa.

Namun untuk menyelenggarakan program pemberian beasiswa sesuai prinsip 3T (Tepat sasaran, Tepat jumlah, dan Tepat waktu) terdapat kendala pada sistem lama di antaranya lamanya proses antara pendaftaran calon penerima beasiswa sampai dengan proses seleksi calon penerima beasiswa. Hal ini sangat sering terjadi ketika terdapat banyak sekali program beasiswa yang memberikan pengumuman secara mendadak dan terkadang banyak mahasiswa yang baru mengetahui info beasiswa di saat mendekati batas akhir pengumpulan berkas sehingga sering terjadi mahasiswa kehilangan peluang untuk mengikuti seleksi beasiswa karena mepetnya waktu untuk menyiapkan syarat-syarat yang diperlukan. Selain itu sering sekali ketika seorang mahasiswa ingin mencoba ikut beberapa seleksi beasiswa sekaligus, terdapat beberapa berkas yang sama yang harus diunggah berkali-kali karena program-program beasiswa yang mahasiswa ikuti memiliki beberapa persyaratan yang sama sehingga masalah seperti ini sebenarnya bisa diakali dengan dibuatkannya sebuah *database* mahasiswa yang menyimpan data-data maupun berkas mahasiswa secara lengkap di mana kedepannya mahasiswa tidak perlu repot-repot lagi ketika membutuhkan data tersebut untuk mendaftar beasiswa.

Adapun permasalahan lain yang terdapat dalam sistem lama yaitu pada proses pengunggahan berkas-berkas mahasiswa berbentuk *file* gambar untuk memenuhi persyaratan pendaftaran beasiswa seperti transkrip, kartu keluarga, surat keterangan miskin, surat keterangan gaji orang tua, dan lain sebagainya. Ketika berkas tersebut diunggah oleh mahasiswa, sistem tidak bisa melakukan validasi data berkas secara otomatis sehingga diperlukan pengecekan secara manual oleh manusia dengan melihat secara langsung *file* yang diunggah oleh mahasiswa. Hal ini tentu saja tidak ada bedanya dengan proses pengumpulan berkas secara konvensional karena masih melibatkan manusia untuk melakukan validasi data berkas padahal seharusnya validasi data berkas tersebut bisa dilakukan oleh sistem.

 Gambar 1. Diagram Alur Sistem Lama

Gambar 1 merupakan gambaran mengenai alur sistem lama. Dari permasalahan itulah, pada Tugas Akhir ini penulis mengusulkan untuk merancang dan membuat sebuah sistem baru yang dapat membantu mempermudah mahasiswa untuk mengikuti seleksi program beasiswa serta membantu kinerja pihak penyeleksi dalam menangani berkas-berkas mahasiswa untuk keperluan seleksi calon penerima beasiswa.

Gambar 2 menunjukkan diagram alur mengenai sistem baru yang akan dibuat.



Gambar 2. Diagram Alur Sistem Baru

Dari sistem validasi berkas otomatis di atas, berikut ini gambaran mengenai detil arsitektur sistem validasi berkas seperti terlihat pada Gambar 3. Dalam arsitektur tersebut terlihat bahwa aplikasi ini akan dibuat di luar SI Beasiswa sehingga diperlukan sebuah *Web Service* untuk menghubungkan antara aplikasi validasi berkas beasiswa dengan aplikasi SI Beasiswa. Sistem validasi ini sengaja dibuat di luar sebuah sistem beasiswa karena dimungkinkan aplikasi validasi berkas beasiswa ini bisa diimplementasikan ke sistem-sistem beasiswa yang lain.

Gambar 3. Arsitektur Sistem Validasi Berkas Beasiswa

Gambar 4 menunjukkan mengenai gambaran *use case* dari sistem validasi berkas persyaratan beasiswa.



Gambar 4. Diagram *Use Case* Sistem Validasi Berkas Beasiswa

Sedangkan untuk berkas-berkas persyaratan pengajuan beasiswa yang dimungkinkan untuk dilakukan validasi data di antaranya adalah sebagai berikut.

* Transkrip
* Kartu Keluarga
* Surat Keterangan Tidak Mampu
* Slip Gaji Orang Tua
* Tagihan Rekening Listrik
* Buku Tabungan
* Dan Lain Sebagainya

**Contoh Pencocokan *String* Dengan Algoritma *Winnowing***

Dalam studi kasus ini misalnya akan dilakukan validasi terhadap sebuah berkas transkrip yang diunggah oleh mahasiswa. Validasi dalam hal ini yaitu melakukan pengecekan apakah berkas yang diunggah tersebut memang benar milik mahasiswa yang bersangkutan. Misalnya mahasiswa tersebut di *database* mempunyai nama “Anak informatika” sedangkan setelah dilakukan ekstraksi gambar transkrip yang diunggah mahasiswa dengan menggunakan OCR didapat teks nama “@anak informatika>”. Secara garis besar untuk melakukan pencocokan string diperlukan konversi setiap karakter yang ada ke dalam bentuk nilai *hash* yang terbentuk dari perhitungan ASCII. Kumpulan-kumpulan nilai *hash* ini disebut dengan *fingerprint* di mana *fingerprint* inilah yang digunakan untuk mencocokan isi teks dari 2 buah dokumen. Berikut ini langkah-langkah penerapan algoritma *Winnowing* dalam pembentukan nilai-nilai *fingerprint* dari tiap dokumen.

1. Melakukan *lowercase* pada setiap karakter dan membuang karakter-karakter dari kedua buah dokumen yang tidak relevan misalnya tanda baca, spasi, dan simbol lain seperti terlihat pada Gambar 5.

Anak informatika

@anak informatika>

anakinformatika

anakinformatika

Gambar 5. Pembersihan Karakter Yang Tidak Relevan Pada Teks Dokumen

1. Mengubah isi dokumen yang telah dilakukan pembersihan ke dalam bentuk rangkaian *gram*, di mana nilai k = 5 seperti terlihat pada Gambar 6.

anakinformatika

anaki infor rmati

nakin nform matik

akinf forma atika

kinfo ormat

Gambar 6. Pembentukan Isi Dokumen Ke Dalam Bentuk Rangkaian *Gram*

1. Melakukan proses *Rolling Hash* untuk menghasilkan nilai *hash* dari setiap *gram* yang terbentuk. Dari proses tersebut didapatkan nilai *hash* dari setiap *gram* seperti terlihat pada Gambar 7.

Berikut ini salah satu contoh perhitungan untuk mendapatkan nilai *hash*.

anaki =

=

= 12126

**12126**, 12917, 12123, 12909, 12840, 13114, 12709, 13457,

13503, 12914, 12352

Gambar 7. Nilai *Hash* Hasil Dari *Rolling Hash*

1. Setelah didapatkan nilai *hash* dari setiap *gram*, selanjutnya yaitu membentuk *window*. Proses pembentukan *window* sama seperti proses *k-gram* dari nilai-nilai *hash* yang dihasilkan dengan besar *window* w = 4 seperti terlihat pada Gambar 8.

{12126, 12917, **12123**, 12909}, {12709, 13457, 13503, 12914},

{12917, 12123, 12909, 12840}, {13457, 13503, 12914, **12352**}

{12123, 12909, 12840, 13114},

{12909, 12840, 13114, **12709**},

{12840, **12114**, 12709, 13457},

{13114, 12709, 13457, 13503},

Gambar 8. Proses Pembentukan *Window*

1. Memilih nilai *hash* terkecil dari setiap *window* untuk dijadikan *fingerprint* dokumen seperti terlihat pada Gambar 9.

**12123, 12709, 12114, 12352**

Gambar 9. *Fingerprint* Dari Sebuah Dokumen

Untuk menghitung persentase kemiripan dari kedua dokumen tersebut digunakan *Jaccard’s Similarity Coefficient* dengan rumus sebagai berikut.

Di mana *fingerprint* pada *database* yaitu a={12123, 12709, 12114, 12352}, dan dokumen transkrip yang diupload yaitu b={12123, 12709, 12114, 12352}. Maka, = {12123, 12709, 12114, 12352} dan = {12123, 12709, 12114, 12352} sehingga,

D(a,b) = 4/4 \* 100 = 100%.

Dari hasil perhitungan di atas dapat ditarik kesimpulan jika nama mahasiswa di *database* sama dengan nama mahasiswa pada berkas transkrip yang diunggah.

# METODOLOGI

1. **Analisis Kebutuhan dan Studi Literatur**

Pada tahap ini dilakukan pengumpulan dan penggalian informasi dari beberapa pihak terkait dengan program pemberian beasiswa dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang akan dibangun. Literatur yang digunakan adalah terkait dengan teknik rekayasa ulang perangkat lunak dan implementasi menggunakan Microsoft ASP .NET Web Application.

1. **Perancangan Sistem**

Pada tahap ini dilakukan analisa awal dan pendefinisian kebutuhan sistem untuk mengetahui masalah yang sedang dihadapi. Dari proses tersebut selanjutnya dirumuskan rancangan sistem yang dapat memberi permecahan masalah tersebut.

1. **Analisis dan desain perangkat lunak**

Gambar 10 menunjukkan desain arsitektur perangkat lunak yang akan dibuat dalam Tugas Akhir.



Gambar 10. Desain Arsitektur Perangkat Lunak

1. **Implementasi perangkat lunak**

Rencana pembuatan perangkat lunak ini akan diimplementasikan dengan menggunakan:

1. Bahasa pemrograman yang digunakan adalah C#.
2. Framework yang digunakan adalah .NET.
3. *Tools* pemrograman yang digunakan adalah Visual Studio 2012.
4. *Tools* pemodelan yang digunakan adalah StarUml dan Power Designer.
5. *Database* yang digunakan adalah SQL Server.
6. **Pengujian dan Evaluasi**

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak yang dibangun dengan menggunakan data yang telah dipersiapkan sebelumnya. Uji coba dan evaluasi perangkat lunak ini digunakan untuk mencari kesalahan program yang timbul, mengevaluasi jalannya program, dan melakukan perbaikan apabila terjadi adanya kesalahan program.

1. **Penyusunan Buku Tugas Akhir**

Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

Tabel 1 menunjukkan rancangan perkiraan jadwal pengerjaan Tugas Akhir.

Tabel 1. Rancangan Jadwal Pengerjaan Tugas Akhir.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2014 | | | | | | | | | | | | | | | | | |
| Januari | | | | Februari | | | |  | Maret | | | | April | | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | “UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 20 TAHUN 2003 TENTANG SISTEM PENDIDIKAN NASIONAL”. [Online]. Tersedia: <http://www.menkokesra.go.id/sites/default/files/file_perundangan/UU%20NO.%2020%20%20-%202003.doc>. [Diakses pada 2 Desember 2013]. |
| [2] | “UNDANG-UNDANG DASAR NEGARA REPUBLIK INDONESIA TAHUN 1945”. [Online]. Tersedia: <http://www.bpk.go.id/web/files/2009/01/UUD45.pdf>. [Diakses pada 2 Desember 2013]. |
| [3] | Depdikbud, *Kamus Besar Bahasa Indonesia*, Jakarta: Balai Pustaka, 1990, hlm. 89. |
| [4] | Soegarda Poerbakawatja, H.A.H. Harahap, *Ensiklopedi Pendidikan*, Jakarta: Gunung Agung, 1982, hlm. 41. |
| [5] | Cornic, P. (2008). Software Plagiarism Detection Using Model-Driven Software Development in Eclipse Platform. [Online]. Tersedia: <http://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/MSc08/Abstracts/CornicPierre-fulltext.pdf>. [Diakses pada 23 Januari 2014] |
| [6] | Creativyst, Inc.”Understanding Classic Soundex Algorithms”. [Online]. Tersedia: <http://www.creativyst.com/Doc/Articles/SoundEx1/SoundEx1.htm>. [Diakses pada 8 Februari 2014] |
| [7] | Karhendana, Arie. *Normalisasi String untuk Optimasi Phonetic String Matching dalam Bahasa Indonesia*. STEI, ITB, 2004. |
| [8] | “Vladimir I. Levenshtein biography”. [Online]. Tersedia: <http://www.ieeeghn.org/wiki/index.php/Vladimir_I._Levenshtein>. [Diakses pada 8 Februari 2014] |
| [9] | Pinzon, Yoan. *Algorithms for Approximate String Matching*. Universidad Nacional de Colombia, 2006. |
| [10] | Sala, Jared. *Longest Common Subsequence*. University New Mexico, 2000. |
| [11] | Cormen, Thomas H., Leiserson, Charles., Rivest, Ronald L., Stein, Clifford. *Introduction to Algorithm*. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1992. |
| [12] | “Optical Character Recognition”. [Online]. Tersedia: <http://searchcontentmanagement.techtarget.com/definition/OCR-optical-character-recognition>. [Diakses pada 23 Januari 2014] |
| [13] | TechTerms,“TechTerms ASP .NET,” [Online]. Tersedia: <http://www.techterms.com/definition/aspnet>. [Diakses pada 23 Januari 2014] |
| [14] | W3C, “Web Services Tutorial”. [Online]. Tersedia: <http://www.w3schools.com/webservices/default.asp>. [Diakses pada 23 Januari 2014] |
|  |  |
|  |  |