

RANCANG BANGUN APLIKASI TEMPLATE SURAT ONLINE DENGAN METODE FUZZY STRING MATCHING

Rio Henri Saputro¹, Dedy Rahman Prehanto²

^{1,2} Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

¹rio.henri1880@gmail.com

²dedyrahman@unesa.ac.id

Abstrak—Pembuatan surat manual sangat berpengaruh terhadap efisiensi serta keseragaman format yang digunakan, untuk beberapa unit bisa dijumpai format surat yang sangat berbeda antara unit satu dengan unit lainnya. Serta waktu yang digunakan untuk membuat surat dari satu pegawai admin sangat panjang, karena disetiap harinya pembuatan surat sangat beragam dari mahasiswa, dosen dan pegawai. Sistem Informasi saat ini sudah sangat berkembang pesat bahkan untuk pembuatan surat dapat dilakukan secara online dan dapat langsung di download untuk membantu mengatasi permasalahan waktu dan ketidak sesuaian format yang digunakan dan memudahkan pengguna untuk mendapatkan layanan administrasi pembuatan surat tanpa harus kesulitan memikirkan format yang digunakan. Tesuo merupakan aplikasi template surat online yang dapat digunakan untuk membuat surat dengan format yang seragam dan tersedia pada aplikasi tanpa harus datang ke kantor administrasi fakultas/unit terlebih dahulu, pengguna langsung dapat mendownload dan mencetak surat dan kemudian membawanya pada kantor administrasi untuk meminta nomor dan tanda tangan. Kemudahan aplikasi ini dibantu dengan pencarian yang menggunakan metode *fuzzy string matching* untuk mempercepat proses pencarian, dimana pengguna dapat lebih mudah mencari surat sesuai dengan kata kunci yang dimasukkan oleh pengguna, ini lebih efisien dan cepat dibandingkan harus mencari menu surat satu persatu tanpa fitur pencarian, serta dilengkapi dengan algoritma *search autocomplete suggestion* untuk memudahkan pengguna mencari surat yang diinginkan dengan hanya menuliskan beberapa huruf dan algoritma akan memprediksi kata apa yang ingin dicari oleh pengguna.

Kata Kunci—Template Surat, Algoritma *Fuzzy String Matching*.

I. PENDAHULUAN

Universitas Negeri Surabaya merupakan lembaga yang kegiatan sehari-harinya tidak terlepas proses surat menyurat sebagai sarana komunikasi seperti pemberitahuan dari Rektorat ke Unit ataupun sebaliknya, serta keperluan perizinan mahasiswa seperti Surat Cuti Kuliah, Tunjangan, Keterangan Masih Kuliah, Ijin Penelitian/Survey (Internal) dan lain sebagainya, serta dari Unesa ke Lembaga lain (Eksternal). Sistem surat menyurat di Unesa sudah dilakukan secara aplikasi, tetapi untuk template surat yang digunakan pada masing-masing Fakultas/Unit terkadang masih berbeda. Contohnya untuk Surat Keputusan

(SK) Mengajar pada Fakultas A berbeda dengan Fakultas B formatnya pasti berbeda-beda antar fakultas, ada yang format lampirannya portrait dan ada pula yang landscape, begitupula dengan format penulisan jam pelajarannya ada yang berbeda. Hal tersebut juga sangat membingungkan ketika dibaca oleh orang diluar fakultas tersebut, begitu pula dengan SK Penguji Skripsi, Penguji PKL, SK Pembimbing Skripsi dan surat keperluan dosen dan tendik lainnya.

Sedangkan untuk surat mahasiswa yang digunakan untuk melakukan keperluan studi di luar kampus masih banyak yang tidak seragam, terkadang pula terdapat mahasiswa dari fakultas yang membuat surat dengan template yang berbeda dengan yang dianjurkan oleh fakultasnya. Terdapat pula fakultas yang sudah menyediakan format pembuatan surat secara offline di kantor tata usaha, dimana mahasiswa dapat membuat surat disana yang kemudian akan dicetak dan dimintakan tanda tangan kepada pejabat yang berwenang dan kemudian diberikan nomor. Ini sangat memangkas waktu mahasiswa karena mereka harus ke kantor Tata Usaha terlebih dahulu untuk membuat surat, Sehingga mahasiswa biasanya terkadang kesulitan untuk membuat surat dari mana saja, kurang informasi mengenai format surat yang digunakan oleh fakultasnya, dan menanyakan kepada teman atau kakak kelas untuk format surat yang digunakan untuk urusan mahasiswa tersebut, akibatnya surat yang dibuat akhirnya berbeda dengan yang seharusnya diterbitkan oleh fakultas. Itu akan sangat kurang baik ketika surat tersebut ditujukan untuk perusahaan atau mitra Unesa yang akan dituju oleh mahasiswa jika surat tersebut berbeda-beda formatnya.

Berdasarkan Permasalahan tersebut perlu dibuat Template Surat yang dapat digunakan untuk semua Fakultas dan Unit agar surat tersebut dapat sama dan seragam antar Fakultas/Unit lainnya. Penelitian ini dilakukan dengan tujuan untuk membuat Aplikasi Template Surat Online yang bertujuan untuk menseragamkan surat dari setiap fakultas dan mampu memberikan kemudahan dalam membuat surat untuk menunjang kegiatan bagi sivitas akademika dan mahasiswa. Mahasiswa pun dapat membuat surat dimanapun dia berada lalu mencetaknya kemudian ke kampus untuk menandatangani surat dan meminta nomor dikantor Tata Usaha Fakultasnya dan untuk bagian persuratan di setiap fakultas dapat membuat surat dengan mudah karena sudah tersedia template sehingga mempercepat

pembuatan surat untuk penunjang kegiatan di masing-masing fakultas.

Dalam aplikasi ini akan terdapat banyak *template* surat yang berbeda-beda, maka selain tombol memilih surat pada aplikasi ini juga akan dilengkapi dengan pencarian surat menggunakan Algoritma Fuzzy *String Matching* dimana pengguna dapat mencari surat sesuai kebutuhan.

II. TINJAUAN PUSTAKA

A. STRING

String adalah tipe data yang berisi text, kalimat, atau sekumpulan karakter. Sebagai contoh, 'A', "ini adalah *String*" atau "Surat no 563" semuanya adalah *String*. Tipe data *string* adalah tipe data yang paling sering digunakan dalam Bahasa pemrograman, dalam PHP *String* memiliki banyak fitur. Karakter yang didukung saat ini adalah 256 karakter ASCII.[1] *String* biasanya dituliskan diantara tanda petik ganda (") sedangkan karakter biasanya dituliskan diantara tanda petik satu ('). Karakter *string* bersifat static dan jika digunakan dalam suatu fungsi, maka nilainya tidak akan berubah selama program aktif.[2]

Dalam PHP sendiri terdapat banyak sekali fungsi untuk *string* seperti `strlen()` untuk mengetahui panjang dari sebuah *string*, `str_word_count()` untuk menghitung jumlah kata dalam satu *string*, `substr()` untuk memotong *string*, `strtoupper()` untuk mengubah semua karakter ke huruf besar, `strtolower()` untuk merubah semua karakter ke huruf kecil, `strrev()` untuk membalik *string*, "stripes" dan "strops" untuk mencari kata secara sensitive case dan insensitive case, dan `str_replace` untuk me-replace suatu karakter dari *string*. [3]

B. Algoritma Pencocokan String

Algoritma Pencocokan *String* ialah algoritma yang digunakan untuk mencari kemunculan suatu *string*, dimana *string* pendek disebut *pattern* dan *string* panjang disebut teks.[6]

String Pendek = *pattern*[0..n-1]

String Panjang = teks[0..m-1]

Pada algoritma ini dapat juga diklasifikasikan berdasarkan arah pencariannya, dimana yang paling umum adalah dari arah kiri ke kanan, seperti ketika membaca kalimat. Dan algoritma beberapa yang termasuk dalam kategori ini ialah[6]:

- Algoritma *Brute Force* dan algoritma *Morris & Pratt*

C. ALGORITMA FUZZY STRING MATCHING

Metode *fuzzy string matching* ialah metode pencarian *string* yang memakai pendekatan terhadap suatu pola dari *string* yang ingin dicari. Melakukan pencarian terhadap *string* yang sama dan atau *string* lain yang mendekati dalam database. Kunci dari metode ini ialah dengan memutuskan bahwa sebuah *string* yang ingin dicari mempunyai persamaan terhadap *string* yang tersimpan dalam sebuah

penampung atau database, walaupun tidak sama persis pola karakternya.[4]

untuk menentukan dan memutuskan tingkat kesamaan dari sebuah *string*, maka diperlukan sebuah "Fungsi Persemaan" (*Similarity function*) dimana fungsi inilah yang akan bertugas untuk menentukan dan memutuskan *string* akhir dari sebuah pencarian *string* melalui hasil pendekatan (Aproksimasi).[5]

Cara kerja dari *fuzzy* ini ialah mencocokkan kemiripan kata dari *pattern* dan text yang di masukkan. Tahap pencocokannya sebagai berikut:

1. String dicocokkan dengan kamus kata yang ada dalam database dan jika memiliki kesamaan dan tidak ada perbedaan maka akan ditampilkan[16]

Contoh : Kamus = KULIAH

Kata yang dicari = KULIAH

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	A	H

2. Apabila belum ditemukan maka di cek kesamaan huruf awal, huruf akhir dan memiliki panjang yang sama, jika ditemukan maka ditampilkan.[16]

Contoh : Kamus = KULIAH

Kata yang dicari = KULIYA

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	Y	A

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	Y	A

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	Y	A

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	Y	A

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	Y	A

0	1	2	3	4	5
K	U	L	I	A	H
K	U	L	I	Y	A

Untuk pencocokan kata diatas dimulai dari index ke-0, jika sama maka dilanjutkan ke i+1 yaitu index ke -2, hingga index ke -3 ditemukan kesamaan dan pada index ke 4 dan ke 5 tidak sama, jika presentase kata yang cocok/sama lebih dari 50% maka ditampilkan

3. Apabila belum ditemukan maka di cek huruf awal dan akhir dengan menhiraukan panjang huruf, apabila ditemukan maka ditampilkan.[16]

Contoh : Kamus = NAIK PANGKAT
Kata yang dicari = PANGKAT

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
P	A	N	G	K	A	T					

Dilakukan pengecekan untuk index ke-0 dari Kamus dan kata yang di cari dan tidak ditemukan kesamaan maka berlanjut ke index selanjutnya.

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
	P	A	N	G	K	A	T				

Pada index ke-2 juga masih belum ditemukan kesamaan, lanjut ke index ke-3 dan seterusnya hingga ditemukan kata cocok pertama

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
		P	A	N	G	K	A	T			

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
			P	A	N	G	K	A	T		

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
				P	A	N	G	K	A	T	

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

Pada index ke-5 ditemukan kata pertama yang cocok kemudian dicek index berikutnya.

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

0	1	2	3	4	5	6	7	8	9	10	11
N	A	I	K		P	A	N	G	K	A	T
					P	A	N	G	K	A	T

Dan ternyata pada index ke-5 sampai index ke 11 ditemukan kata yang mirip maka kata ditampilkan

4. Apabila masih belum ditemukan atau tidak ada kemiripan dengan kamus data maka ditampilkan sesuai dengan kata yang dimasukkan pengguna. Dalam aplikasi ini akan dimunculkan pesan “surat tidak ditemukan”[16]

D. FUZZY TIME SERIES

Fuzzy time series adalah suatu metode prediksi data yang memakai prinsip-prinsip *fuzzy* yang dasarnya adalah menangkap rumus data lama lalu kemudian memproyeksikan hasil data baru. Model prediksi dengan menggunakan metode *fuzzy time series* memiliki beberapa tahapan sebagai berikut[15]:

1. Definisi sebagai *universal U* pada himpunan *fuzzy* sehingga didapat $U=[x,y]$
2. Menentukan Skor Minimal dan maksimal untuk data riwayat actual ($X_{min} = x, X_{max} = y$).
3. membagi *universal U* dengan beberapa rangkaian data u_1, u_2, \dots, u_n dan kemudian membentuk skor linguistik.
4. melakukan Fuzzifikasi berdasarkan kumpulan data *fuzzy* histori aktual.
5. Menghitung Skor *Fuzzy* histori aktual dengan menggunakan pola:

$$\mu_{NB}[x] = \frac{b-x}{b-a} \quad 203659 \leq x \leq 471585$$

6. Memilih basis model W yang sesuai untuk menghitung *fuzzy* dengan pola:

$$r(MBF)^{w+1} = \frac{MBF_1 + MBF_2 + \dots + MBF_w}{w}$$

7. Melakukan defuzzifikasi hasil perhitungan dari langkah diatas menggunakan pola:

$$y = \frac{(A1 * c1 + A2 * c2)}{A1 + A2}$$

$$v = r(NB) * A1 + r(NM) * A2 + \dots + r(PB) * A7$$

$$Fi = y + v$$

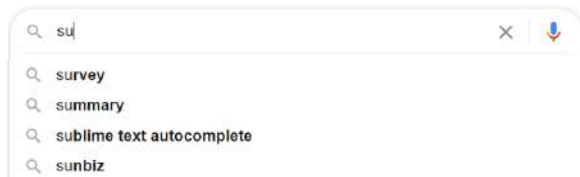
[15]

E. SEARCH AUTOCOMplete

Search Autocomplete adalah algoritma yang berisi prediksi *query* yang menawarkan saran kata yang sering dicari atau sedang ingin dicari, sehingga mempercepat interaksi antara manusia dan computer dan membuat pengguna lebih mudah untuk menyelesaikan *query*-nya yang sangat berguna bagi individu yang kurang paham teknis kata yang sedang mereka cari, dan statistiknya dapat di kumpulkan pada tempat penampung data (*database*) [8]

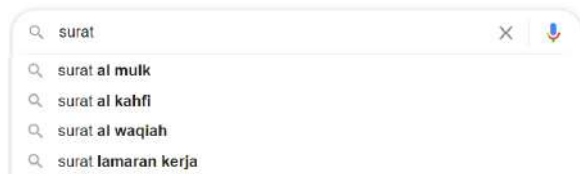
Layanan *Autocomplete* biasanya terdapat pada mesin pencari seperti google, bing dll. Kegunaan fitur seperti ini dibuat berdasarkan prediksi algoritma yang mengidentifikasi kata yang sesuai dari *query* atau *database*. Mesin pencari akan memilih saran kata yang sesuai dengan *query* yang terkait. Cara kerjanya ialah dengan menuliskan sebuah/beberapa huruf awal dan program akan otomatis mencari kata relevan dengan huruf/kata awal tersebut.[8]

Sebagai contoh pada layanan mesin pencari di google dengan menuliskan beberapa kata dapat memprediksi kata selanjutnya yang mungkin sesuai seperti pada gambar 1.



Gambar 1. Contoh prediksi berdasarkan beberapa huruf pertama.

Pada pencarian huruf prediksi kata mengacupada kesesuaian dari 2 huruf pertama, dan hasilnya akan berbeda-beda dan biasanya kurang sesuai dengan yang pengguna inginkan seperti terlihat pada gambar 1 dimana kata yang terprediksi ialah "survey, summary, sublime, dan sunbiz" dan dari 4 kata tersebut sangat jauh sekali persamaannya, berbeda jika pengguna mencoba mencari dengan menuliskan kata pertama seperti pada gambar 2.



Gambar 2. Contoh prediksi berdasarkan kata pertama

Terlihat bahwa hasil pencarian dengan kata seperti "surat" lebih spesifik mengacu pada kesesuaian hasil yang tidak terlalu melenceng jauh dari kata "surat".

F. PHP

PHP adalah kependekan dari Hypertext PreProcessor adalah Bahasa program *script* sederhana yang digunakan untuk memproses HTML Form atau konten HTML di dalam web (Web Page).[9] PHP sendiri dibuat di tahun 1995 oleh Rasmus Lerdorf dan dikembangkan menjadi PHP3 di tahun 1998 oleh komunitas pengguna PHP dimana PHP3 saat itu memperkenalkan penggunaan *object* dan *class*. Kemudian di tahun 2000 dirilis PHP4 yang juga masih menerapkan konsep *object oriented* yang masih sangat terbatas.[9]

Script PHP sangat mudah dipelajari dan sering digunakan untuk membuat Website Dinamis. Kemudahan dan Kesederhanaan PHP sangat membantu pengguna baru yang bahkan tidak mengerti sama sekali dengan PHP bisa dengan cepat dipelajari. PHP merupakan bahasa open source (Gratis) yang tidak memerlukan pembelian lisensi dan dapat digunakan diberbagai sistem operasi seperti Linux, Unix, Macintosh dan Windows.[10]

G. SURAT

Surat adalah sehelai atau lebih kertas yang digunakan sebagai alat komunikasi untuk memberi data atau informasi seperti pemberitahuan, pernyataan, permintaan, laporan dan lain sebagainya secara tertulis antara dua belah pihak atau lebih.[7] Surat sendiri dibagi berdasarkan sifat suratnya

seperti surat pemberitahuan terbagi dua yaitu pemberitahuan terbatas atau pemberitahuan umum, untuk pemberitahuan terbatas biasanya hanya diberikan kepada beberapa orang saja dan lebih bersifat pribadi atau rahasia dan pada kolom perihal biasanya tertulis "Terbatas, Rahasia, atau Sangat Rahasia". Sedangkan untuk surat pemberitahuan umum biasanya berisi tentang informasi yang dapat dikonsumsi atau diterima oleh banyak orang dan tidak bersifat pribadi atau rahasia, pada kolom perihal biasanya tertulis "Pemberitahuan", contohnya surat pemberitahuan tentang hari libur nasional, pemberitahuan pengisian pajak, atau pemberitahuan pelaksanaan kegiatan.

Untuk Surat Pernyataan biasanya bersifat informatif dan berisi data yang lebih spesifik mengenai sesuatu seperti Nama, alamat, nomor telfon, jenis kelamin, tempat tanggal lahir, serta pernyataan jelas oleh yang menyatakan, biasanya surat pernyataan ini lebih bersifat pribadi atau personal dan terkadang juga terdapat materai untuk pengesahan pernyataan tersebut. surat ini biasanya tidak terdapat kolom perihal pada bagian atas surat karena bersifat pribadi atau personal.

Surat Permintaan biasanya berisi tentang suatu permohonan atau permintaan dari suatu unit, organisasi atau lembaga, biasanya berupa permintaan narasumber/pembicara, permintaan data mahasiswa atau pegawai dan data lain yang berhubungan dengan suatu kepentingan yang bersifat pribadi maupun umum. Pada kolom perihal biasanya ditulis Permintaan.

H. LARAVEL

Laravel adalah sebuah *framework* PHP yang memiliki lisensi MIT dan menggunakan konsep *model, view, dan controller* atau sering disingkat MVC yang dikembangkan oleh Taylor Otwell, didalam laravel sudah tersedia sintaks-sintaks yang sangat jelas, ekspresif dan sangat mudah digunakan sehingga dapat menghemat waktu untuk pembuatan suatu aplikasi website. Dan aplikasi ini bersifat open source (Gratis) dan tidak membutuhkan pembelian lisensi.[11] konsep MVC pada laravel membagi pembuatan aplikasi website dengan masing-masing tugas dan tanggung jawab, dimana model bertugas untuk mengatur, menyiapkan dan memanipulasi data atas perintah dan instruksi dari *controller*, sedangkan *view* bertugas untuk menyajikan informasi data yang diinstruksikan oleh *controller*, dan *Controller* sendiri bertugas untuk mengatur data apa saja yang harus dikelola oleh model dan ditampilkan oleh view atas perintah pengguna.[11]

I. BASIS DATA (DATABASE)

Basis Data adalah kumpulan data atau informasi yang tersimpan dalam tabel yang saling berhubungan (berelasi) yang terdapat pada penyimpanan elektronik. Dapat disimpulkan bahwa masing-masing tabel digunakan untuk menyimpan data dan dari masing-masing tabel data tersebut akan saling berhubungan. Data tersebut dapat diakses menggunakan perangkat lunak komputer dengan menggunakan fitur kueri (*query*)[12]

J. POSTGRESQL

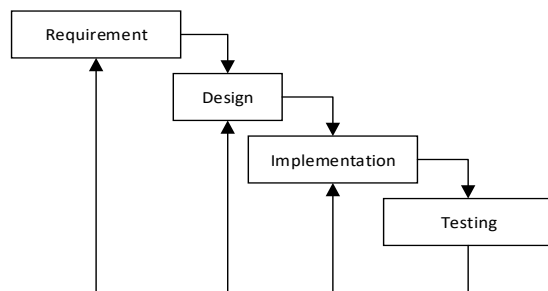
PostgreSQL ialah perangkat lunak (*Software*) sistem basis data yang disebarluaskan secara bebas dengan lisensi BSD. *Software* ini menyediakan banyak fitur yang sangat membantu memudahkan pengolahan data seperti DB Mirror, PGCluster, PGPool dan lain sebagainya. *Software* ini bahkan dapat menyimpan data berupa objek geometric seperti garis (*line*), titik (*point*), dan area (*polygon*) dan lainnya dengan menggunakan tambahan plugin PostGIS.[13]

K. SISTEM INFORMASI

Pengertian yang paling sederhana dari sistem informasi adalah sistem yang memiliki *input*, proses, dan *output*. Dimana sistem informasi merupakan suatu cara menyampaikan informasi kepada pengguna berdasarkan data/informasi yang telah diolah sedemikian rupa hingga menghasilkan informasi yang diinginkan.[14] Tujuan sistem informasi adalah menyediakan data yang akan digunakan untuk perencanaan, pengamatan, perhitungan dan pengambilan keputusan.

III. METODOLOGI PENELITIAN

Berikut ini adalah alur untuk penelitian “Rancang bangun aplikasi template surat online dengan metode *fuzzy string matching*” dapat dilihat pada gambar 2.



Gambar 2. Alur Penelitian

Pada Gambar 2 dapat dilihat bahwa tahapan pertama yang dilakukan dalam penelitian ini yaitu *Requirement* yaitu menganalisis kebutuhan dengan studi pustaka dan mencari pustaka yang sesuai dan relevan agar dapat menjadi referensi pendukung yang baik. Pustaka yang diambil ialah pustaka yang mengacu pada Metode *Fuzzy String Matching*. Kemudian tahap kedua *Design* yaitu mendesain database aplikasi Pencarian Surat dengan *PostgreSQL* menggunakan bantuan aplikasi *Navicat*. Dan Pada tahap *implementation* ini menggunakan aplikasi *Laravel* dan penulisannya menggunakan bahasa PHP (*Hypertext Preprocessor*). dan yang terakhir adalah tahap pengujian (*Testing*), dimana pada tahap pengujian ini dilakukan dengan 6 pengguna untuk melakukan pencarian dan pembuatan surat dengan menggunakan *Web Browser Google Chrome*.

A. Pengumpulan Data

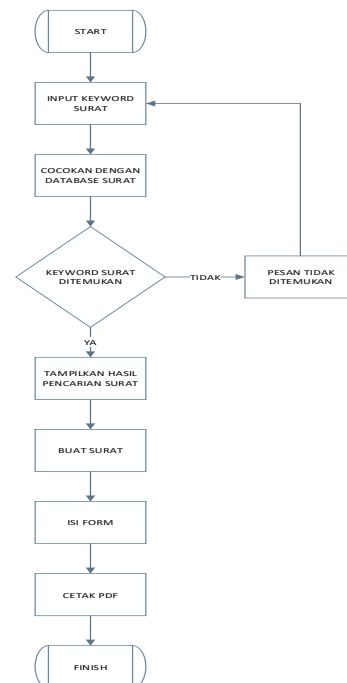
Pada tahap ini penulis melakukan studi pustaka yang terkait dengan teori pada algoritma *Fuzzy String Matching*

yang ditekankan pada pencarian surat guna mempermudah pencocokan pencarian surat dengan keyword surat yang ada di database. penulis menggunakan template surat yang ada pada Universitas Negeri Surabaya untuk mempermudah pembuatan surat, sehingga tidak perlu datang ke Kampus atau kantor administrasi Fakultas dan jurusan untuk pembuatan suratnya.

Pada penelitian ini penulis memasukkan keyword yang sesuai pada setiap database surat sehingga mempermudah pencarian surat yang akan dibuat, keyword dibuat berdasarkan subjek kumpulan dari kategori surat (mahasiswa, dosen, tendik, kepegawaian, pemberitahuan, dll), judul surat, dan peruntukannya.

Untuk Format suratnya sendiri menggunakan aturan persuratan yang ada di Universitas Negeri Surabaya yang diseragamkan sehingga semua Fakultas, Jurusan dan Unit kerja mempunyai satu format yang seragam dan tidak berbeda-beda seperti sebelumnya.

B. Desain Sistem



Gambar 3. Alur Desain Sistem

Pada Gambar 3. Menampilkan desain alur pada untuk membuat surat dengan melakukan pencarian terlebih dahulu menggunakan metode *Fuzzy String Matching* secara umum. Untuk membuat surat menggunakan aplikasi berbasis web yang dapat dibuka menggunakan *web browser*.

C. Rancangan Aplikasi

Penelitian ini mengimplementasikan metode *Fuzzy String Matching* pada menu pencarian surat sehingga dapat mempermudah pengguna untuk mencari dan menemukan surat yang dimaksud dengan waktu yang singkat, dibandingkan harus mencari judul surat satu per satu

ditengah banyaknya macam surat yang ada pada aplikasi sehingga memperlambat waktu pengguna. dan pembuatan aplikasinya sendiri menggunakan aplikasi Laravel dan database PostgreSQL.

D. Ujicoba

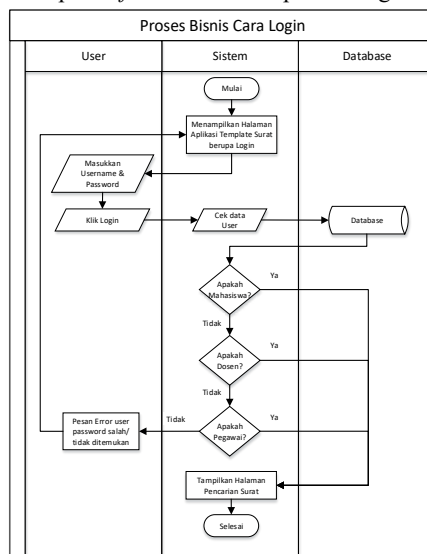
Ujicoba untuk penelitian ini dilakukan oleh 3 pengguna yang terdiri dari 1 mahasiswa, 1 dosen, dan 1 pegawai administrasi yang akan mencoba mencari template surat dengan menu pencarian yang memanfaatkan metode *Fuzzy String Matching* untuk pencariannya, sehingga dapat dilihat keefektifan metode ini untuk mencari surat yang spesifik sesuai yang diinginkan dalam aplikasi, kemudian menggunakan template yang dicari untuk membuat surat secara online dan langsung dapat mengunduh suratnya dalam format .pdf yang tersedia pada aplikasi.

IV. HASIL DAN PEMBAHASAN

Pada Bab ini berisi tentang hasil pengujian terhadap implementasi yang dilakukan serta analisisnya. Melakukan pengujian merupakan cara untuk menilai tingkat kesesuaian hasil pencarian dengan metode *Fuzzy String Matching* dan mrlisht apakah metode pencarian autocomplete dapat menemukan kesesuaian dengan kata yang dimaksud jika kata tersebut salah penulisan atau kurang.

A. Flowchart

Berikut merupakan *flowchart* untuk proses Login



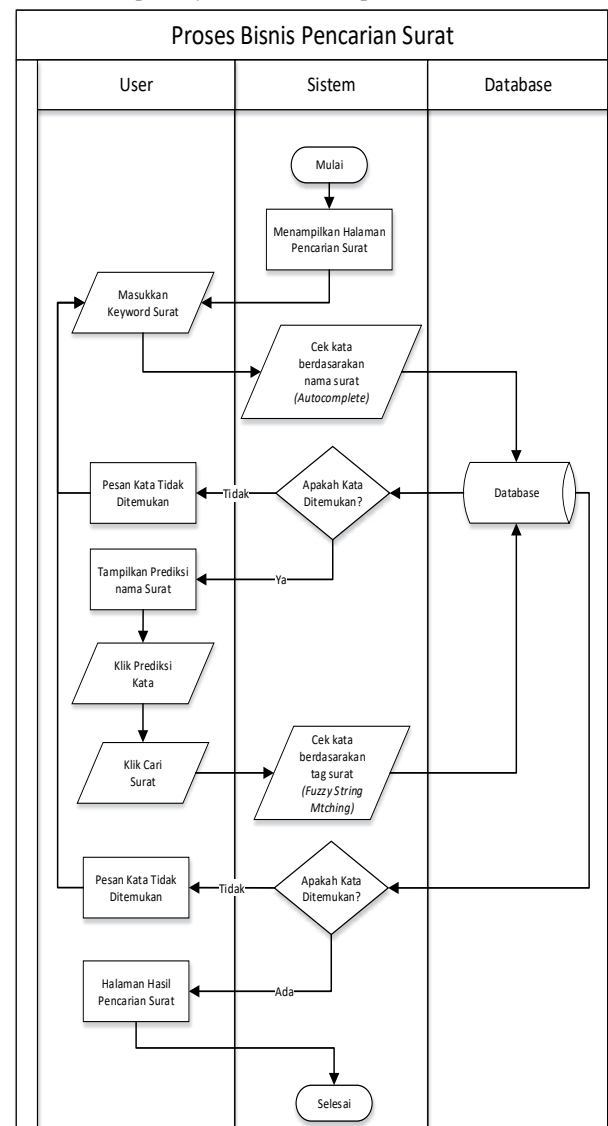
Gambar 4. *Flowchart* Proses Login

Untuk *user* login berdasarkan email masing-masing individu. Ketika diinputkan emailnya maka sistem akan memproses pengecekan *user* tersebut di database berdasarkan hak aksesnya apakah Mahasiswa, Dosen, atau pegawai. Jika berhasil maka akan tampil halaman awal pencarian surat berdasarkan hak aksesnya masing-masing. Jika mahasiswa maka jenis surat yang dapat dicari hanya terbatas surat untuk keperluan mahasiswa seperti Surat Keterangan Aktif Kuliah, Kehilangan, Penurunan UKT,

Kunjungan, PKL/Magang, Observasi Penelitian, Observasi Pengambilan Data, dan lainnya.

Kemudian untuk Dosen hak akses suratnya hanya terbatas untuk keperluan dosen seperti Surat Pengesahan PKM, Surat Pengajuan dan pernyataan Penelitian, usulan studi lanjut, usulan kenaikan pangkat, satyalencana dan pensiun dan surat-surat lain yang berhubungan dengan keperluan dosen. untuk pegawai hak aksesnya tidak terbatas untuk mahasiswa saja tetapi juga dosen, semua surat yang berhubungan dengan mahasiswa dan dosen bisa di akses oleh pegawai jurusan, fakultas, maupun pusat.

Berikut merupakan *flowchart* untuk proses Pencarian Surat



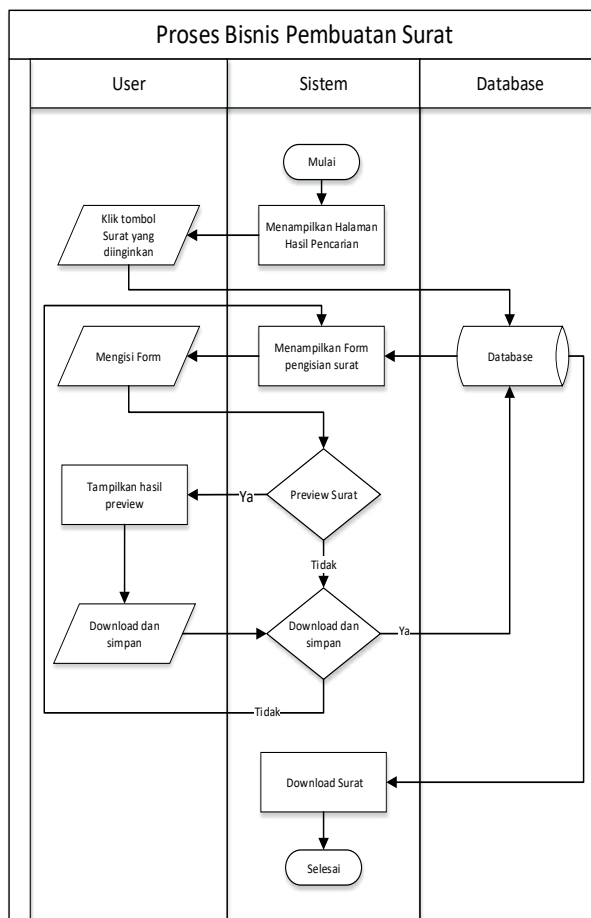
Gambar 5. *Flowchart* Proses Pencarian Surat

Untuk Proses Pencarian akan dimulai dengan langkah-langkah sebagai berikut:

1. Setelah login berhasil, maka sistem akan mulai menampilkan halaman Pencarian, dan kemudian pengguna akan memasukkan kata kunci surat yang ingin dicari pada kolom pencarian.

2. Kata kunci akan diperiksa oleh algoritma *autocomplete* berdasarkan nama surat dalam database. jika tidak di temukana maka akan muncul pesan “Tidak Ditemukan Data”, jika ditemukana maka akan muncul prediksi kata di bawah kolom pencarian yang ada sehingga pengguna dapat langsung memilih kata tersebut atau tidak. jika ingin memilih maka pengguna dapat mengklik prediksi kata tersebut, jika tidak pengguna dapat langsung menekan tombol “Cari Surat” yang ada.
3. Kemudian setelah pengguna akan menekan tombol “Cari Surat” dan kata kunci yang dimasukkan akan di cek dengan algoritma *Fuzzy String Matching* berdasarkan tag surat yang ada pada database. Jika tidak ditemukan maka akan muncul pesan “Surat Tidak Ditemukan” dan akan otomatis kembali ke halaman pencarian surat, jika ditemukan maka akan muncul halaman hasil pencarian surat yang berisi surat-surat yang relevan dengan kata pencarian yang ditulis oleh pengguna.

Berikut merupakan *flowchart* untuk proses Pembuatan Surat



Gambar 6. *Flowchart* Proses Pembuatan Surat

1. Setelah muncul halaman hasil pencarian yang diinginkan oleh pengguna, maka pengguna dapat memilih surat yang diinginkan dan akan diarahkan

untuk pengisian form surat berdasarkan surat yang dipilih.

2. Data pengguna diambilkan dari biodata pengguna dalam database sehingga pengguna hanya perlu mengisi data lain yang tidak berhubungan dengan biodata pengguna, prodi dan fakultas.
3. Setelah selesai mengisi form sederhana surat dapat di preview dan jika sudah sesuai makan tinggal di download.
4. Hasil download akan berupa file .pdf

B. Implementasi Database

Tabel 1. Tabel Nama Surat

Nama Surat		
id	int8	64
nama_surat	varchar	255
tag	text	0
jenis_surat	varchar	255
link	text	0
pemilik	varchar	255

Tabel 2. Tabel Surat Keterangan Aktif Kuliah Mahasiswa

Surat Keterangan Aktif Kuliah		
id	int8	64
kop_id	varchar	255
kode_nomor	varchar	255
nama_mhs	varchar	255
nim_mhs	varchar	11
prodi_mhs	varchar	255
nama_fak	varchar	255
semester	int8	64
ajaran	int8	64
nama_pndtngn	varchar	255
nip_pndtngn	varchar	18
jabatan_pndtngn	varchar	255
date	date	0

Tabel 3. Tabel Surat Pengesahan PKM 4 Orang

Surat Pengesahan PKM 4 Orang		
id	int8	64
kop_id	varchar	255
judul_penelitian	text	0
nama	varchar	255
nip	varchar	18
prodi	varchar	255
instansi	varchar	255
no_hp	varchar	13
email	varchar	255
bidang	varchar	255
alamat_kantor	text	0

nama_anggota_i	varchar	255
nama_anggota_ii	varchar	255
nama_anggota_iii	varchar	255
lokasi_wilayah	varchar	255
kabupaten	varchar	255
propinsi	varchar	255
jangka_waktu	varchar	255
biaya	int8	64
date	date	0
nama_pndtngn_pejabaat	varchar	255
nip_pndtngn_pejabaat	varchar	18
jabatan_pndtngn_pejabaat	varchar	255
nama_pndtngn_pengesah	varchar	255
nip_pndtngn_pengesah	varchar	18
jabatan_pndtngn_pengesah	varchar	255

C. Implementasi Sistem

Pada Penelitian ini pengguna ialah mahasiswa, dosen, dan pegawai yang sudah memiliki email individu yang berisikan data pribadi yang kemudian akan dijadikan sebagai *user* untuk login dalam aplikasi. Untuk surat juga berbeda-beda yang dapat diakses oleh masing-masing *user*, *user* mahasiswa hanya bisa mencari surat yang berhubungan dengan mahasiswa, *user* dosen hanya surat yang berhubungan dengan dosen saja, dan untuk pegawai dapat mengakses surat untuk dosen dan mahasiswa. Berikut adalah tampilan halaman awal login:



Gambar 7. Tampilan login Awal

Percobaan pada Gambar 7 menggunakan akun mahasiswa untuk login. Setelah berhasil login akan tampil halaman pencarian dimana pengguna dapat mengisi *keyword* atau kata kunci yang diinginkan untuk mencari surat. Berikut tampilan halaman pencarian seperti ditunjukkan pada Gambar 8:



Gambar 8. Halaman Pencarian

Sistem pencarian menggunakan algoritma *Fuzzy String Matching* dan fitur *Autocomplete* yang akan mempermudah

pencarian format surat yang diinginkan dari banyaknya format surat yang ada. Berikut contoh pengaplikasian fitur *autocomplete* untuk pencarian:



Gambar 9. Pencarian menggunakan keyword Kuliah

Ketika pengguna memasukkan kata kunci “Kuliah” maka sistem akan memproses kata tersebut dan mencari dalam database nama surat, apakah terdapat kata “kuliah” dan jika ada maka akan langsung ditampilkan dalam kolom dibawah kolom inputan seperti pada gambar 9.

Kemudian pengguna dapat memilih kata prediksi tersebut dan kemudian klik “cari surat” dan sistem akan memproses kata tersebut dalam database dan mencari “tag” pada nama surat sesuai kata yang dicari. Berikut halaman hasil pencarian untuk kata “Aktif Kuliah”:



Gambar 10. Hasil Pencarian surat dengan *keyword* “aktif kuliah”

Jika pengguna (mahasiswa) tersebut mencoba mencari surat yang berhubungan dengan dosen seperti kepankangan/ijin belajar dan lain sebagainya maka tidak akan muncul karena hak aksesnya berbeda dan halaman pencarian awal fitur *autocomplete* akan memunculkan pesan “Kata Tidak Ditemukan” seperti pada gambar berikut:



Gambar 10. Pencarian Surat dengan *keyword* yang tidak ditemukan

Dan hasil halaman pencarian juga akan menampilkan pesan sebagai berikut dan akan kembali ke halaman pencarian awal setelah beberapa detik



Gambar 11. Halaman Pencarian Surat tidak ditemukan

D. Pengujian Sistem

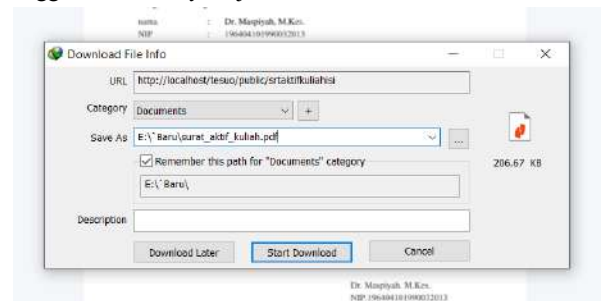
Pengujian akan dilakukan oleh mahasiswa dosen dan pegawai dimana *user* mahasiswa akan membuat surat keterangan aktif kuliah, dosen akan membuat surat Pengesahan PKM pegawai akan mencoba mencari surat milik mahasiswa dan surat milik dosen, karena hak akses pegawai dapat membuat semua surat.

Gambar 12. Form pengisian surat keterangan aktif kuliah mahasiswa

Pada gambar diatas *user* mahasiswa tidak perlu mengisi Data akademis seperti nama, nim, prodi, karena akan diambilkan dari data *user* dan sistem akan otomatis menggunakan data pribadi *user*, Kecuali untuk beberapa surat yang bersifat kelompok seperti Kunjungan PKL, PI/Magang maka data teman dapat dituliskan secara manual pada form pengisian. Kemudian untuk surat keterangan aktif kuliah ini data penandatangan juga tidak perlu diisi karena sistem akan melihat data prodi mana *user* berasal dan akan otomatis mengisi data pejabat berwenang yang menandatangani surat keterangan aktif kuliah pada fakultas dan prodi, dimana sudah pasti adalah dekan. Berdeda lagi jika *user* membuat surat keterangan ujian skripsi, disitu pengguna akan diminta untuk mengisi data nama dan NIP dari pembimbing dan penguji skripsi pengguna secara manual.

Untuk surat keteranga aktif kuliah ini Yang perlu diisi hanya data semester berapa dan tahun. Setelah itu *user* dapat melihat hasil surat dengan tekan tombol "preview"

Gambar 13. Hasil surat Keterangan aktif kuliah mahasiswa Setelah dirasa sudah sesuai maka *user* dapat langsung mengunduh data surat tersebut dalam format .Pdf dan tinggal mencetaknya saja.



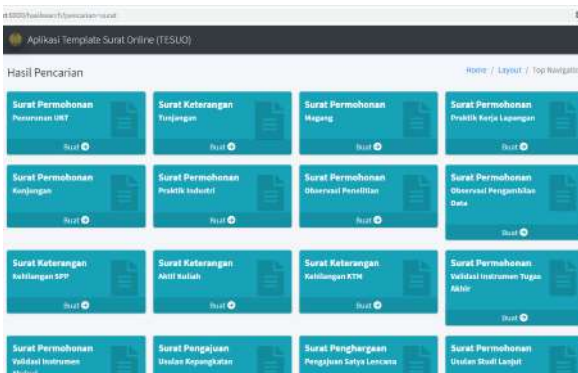
Gambar 14. Proses download surat yang dibuat

Pada *user* dosen juga tidak perlu mengisi data pribadi dan hanya perlu mengisi data judul PKM, nama anggota, lokasi penelitian, jangka waktu, biaya, pejabat yang mengetahui dan pejabat yang mengesahkan saja.

Gambar 15. Form pengisian surat Pengesahan PKM Dosen



Gambar 16. Hasil Preview surat Pengesahan PKM Dosen
Kemudian dapat di unduh dengan menekan tombol
“download”. Percobaan selanjutnya menggunakan *user*
pegawai mencoba mencari surat mahasiswa dan surat
dosen, apakah dapat muncul atau tidak seperti pada gambar
berikut:



Gambar 17. Hasil Pencarian dengan kata kunci surat
menggunakan *user* pegawai

Pada *user* pegawai jika di cari dengan kata kunci “surat”
maka semua surat yang berhubungan dengan mahasiswa,
pegawai dan dosen akan muncul semua.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Untuk fitur *autocomplete* akan langsung membaca data yang
dituliskan pada kolom pencarian dan akan langsung
mencocokkan pada database apakah terdapat kata yang
sesuai atau tidak, jika ada maka akan muncul kolom berisi
data yang dicari dibawah kolom pencarian, jika tidak ada
maka akan muncul kolom berisi “kata tidak ditemukan”.
Setelah ditekan “cari”, maka algoritma *Fuzzy String*
Matching akan mencocokkan kata yang dicari dengan
database yang akan dan memunculkan hasil yang dicari,
dan jika tidak ditemukan hasil maka akan langsung

kembali ke halaman pencari. dengan demikian penggunaan
fitur *Autocomplete* dan algoritma *fuzzy* ini sangat
membantu mempermudah pengguna untuk menemukan
surat yang diinginkan dengan cepat dan menghemat waktu
pencarian. untuk pencarian juga dibatasi berdasarkan hak
akses sebagai mahasiswa, dosen atau pegawai, dan untuk
pembuatan surat pengguna tidak perlu menuliskan data
informasi akademis dan pribadi, serta untuk surat tertentu
pengguna juga tidak perlu mengisi data penandatanganan
karena otomatis akan diisi sesuai pejabat berwenang yang
menandatangani.

B. Saran

Untuk penelitian selanjutnya mungkin dapat ditambahkan
fitur lain yang dapat membantu mempermudah pengguna
mencari dan menghemat waktu pengguna untuk membuat
surat tersebut.

REFERENSI

- [1] Andre, 2014. “Tutorial Belajar PHP Part 15: Mengenal Tipe Data String dan Cara Penulisan String dalam PHP” [Online]. Available: <https://www.duniailkom.com/mengenal-tipe-data-string-dan-cara-penulisan-string-dalam-php/>. Diakses tanggal 3 Maret 2020
- [2] Kusuma, Widia Rahmawati, 2019. “Apa itu variable dan Fungsi String?” [Online]. Available: <https://redaksi.pens.ac.id/2019/10/24/apa-itu-variabel-dan-fungsi-string/>. Diakses pada 3 Maret 2020
- [3] Huda, Nurul, 2020. “PHP Dasar: Manipulasi String” [Online]. Available: <https://jagongoding.com/web/php/dasar/manipulasi-string/>. Diakses tanggal 3 Maret 2020
- [4] Gurning A. I. A., Zarnelly, Adawiyah A. 2016, “Penerapan Fuzzy String Matching pada Aplikasi Pencarian Tugas Akhir Mahasiswa Jurusan Sistem Informasi Berbasis Web (Studi Kasus: Fakultas Sains dan Teknologi UIN Suska Riau)”. Jurnal Rekayasa dan Manajemen Sistem Informasi, Vol. 2, №1, Februari 2016, 54- 59
- [5] Tregoat, Julien, 2018. “An Introduction to Fuzzy String Matching”. [Online]. Available: <https://medium.com/@julientregoat/an-introduction-to-fuzzy-string-matching-178805cca2ab>. Diakses tanggal 12 Maret 2020
- [6] Dirgandhavi, Naufal Aditya, 2019. “Algoritma *Fuzzy String Matching* untuk Pencocokan String”. Makalah IF2211 Strategi Algoritma, Semester II Tahun 2018/2019.
- [7] Ali, Adlan, Tanzili, 2006. “Pedoman Lengkap Menulis Surat”, PT Kawan Pustaka, Hal 1
- [8] Wang, Peng, dkk, 2018. “Game of Missuggestions: Semantic Analysis of Search-Autocomplete Manipulations”. Network and Distributed Systems Security (NDSS) Symposium 2018. 18-21 February. ISBN: 1-1891562-49-5.
- [9] Aziz, M Farid, 2005. “Object Oriented Programming dengan PHP5, Panduan Praktis Pengembangan OOP dengan PHP5, mengimplementasikan UML dan Patterns”, Elex Media Komputindo. Hal 1.

- [10] Anhar, 2010. “Panduan menguasai PHP & Mysql Secara Otodidak”, Mediakita, Hal 3.
- [11] Yudhanto Yudho, Prasetyo Helmi Adi, 2019. “Mudah Menguasai Framework Laravel”. Elex Media Computindo, Hal 22-29.
- [12] Dewi, Aliftika Cipta, 2019. “Artikel Ilmiah Konsep Basis Data Relasional”
- [13] Savira, Donna, 2020. “Sistem Basis Data Artikel Pengenalan Database Berbayar PostgreSQL” [Online]. Available: <https://ilearning.me/2020/02/24/artikel-sistem-basis-data-postgresql/>. Diakses Tanggal 25 April 2020
- [14] Prehanto, Dedy Rahman, 2020. “Library Book Modeling Data Using the Association Rule Method with Apriori Algorithm in determining Book Placement and Analysis of Book Loans”. International Journal of Advanced Science and Technology, Vol. 9, No. 5, (2020). Pp 1244-1250
- [15] Prehanto, Dedy Rahman, 2019. “Soil Moisture Prediction using Fuzzy Time Series and Moisture Sensor Technology on Shallot Farming”. The 4th International Conference on Energy, Environment, Epidemiology and Information System (ICENIS 2019). Volume 125 number 23002.
- [16] Resid Rahmayanto, Abror, 2013. “Perancangan *Intelegent Agent Chatbot* Dengan Menggunakan Metode Fuzzy String Matching”. Laporan Tugas Akhir. Bab III point 3.8.2.