# MEMBANGUN SISTEM REPOSITORY BERBASIS SEARCH ENGINE MENGGUNAKAN ALGORITMA KNUTH MORRIS PRATT

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan Pendidikan Jenjang Sarjana Terapan**

**Pada Politeknik Negeri Lhokseumawe**



**Oleh SAID MAHFUD**

**NIM 1490343096**

**Program Studi : Teknologi Rekayasa Komputer Jaringan Jurusan : Teknologi Informasi dan Komputer**

**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI POLITEKNIK NEGERI LHOKSEUMAWE**

**2018**





iii

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk mendapat gelar kesarjanaan di suatu Perguruan Tinggi, dan sepengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebut dalam daftar pustaka.

Buketrata, 12 Juli 2018

Said Mahfud

iv

Segala puji beserta syukur senantiasa penulis panjatkan kepada Allah SWT yang telah memberikan rahmat dan petunjuk-Nya kepada penulis sehingga dapat menyelesaikan tugas akhir yang berjudul Membangun Sistem *Repository* Berbasis *Search Engine* Menggunakan Algoritma *Knuth Morris Pratt*. Shalawat beserta salam kepada baginda Rasulullah Muhammad SAW, keluarga tercinta serta para sahabat yang telah berjasa dalam memperjuangkan islam dan ilmu pengetahuan. Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan Sarjana Terapan Program Studi Teknologi Rekayasa Komputer Jaringan Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe.

Dalam penyusunan tugas akhir ini penulis banyak mendapat saran, dorongan, bimbingan dan masukan serta keterangan-keterangan dari berbagai pihak sehingga menjadikan hal ini pengalaman yang tidak dapat diukur secara materi, namun dapat membukakan pikiran dan mata penulis bahwa sesungguhnya pengalaman dan pengetahuan tersebut adalah dosen yang terbaik bagi penulis. Oleh karena itu, penulis mengucapkan terimakasih kepada Bapak Anwar, S.Si, M.Cs dan Bapak Muhammad Nasir S.T., M.T. selaku pembimbing I dan II yang telah membimbing penulis dengan tulus dan ikhlas dalam penyelesaian tugas akhir ini. Ucapan terimakasih penulis sampaikan kepada Ibu Mursyidah S.T., M.T. Ketua Program Studi Teknologi Rekayasa Komputer Jaringan, Bapak Muhammad Arhami, S.Si., M.Kom Ketua Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe, seluruh staff pengajar Program Studi Teknologi

v

Rekayasa Komputer Jaringan, serta seluruh staff administrasi pada Jurusan Teknologi Informasi dan Komputer. Selanjutnya, penulis juga tidak lupa mengucapkan terima kasih kepada sahabat dan teman-teman se-angkatan Teknologi Rekayasa Komputer Jaringan angkatan 2014 dan semua pihak yang telah berpartisipasi dalam penyelesaian tugas akhir ini. Teristimewa kepada Ayahanda Said Nazaruddin dan Ibunda Cut Meutia yang selalu memberikan dukungan dan doanya sehingga dapat menyelesaikan tugas akhir ini.

Semoga tugas akhir ini dapat bermanfaat bagi diri pribadi penulis dan juga bagi pembaca nantinya. Amiin ya Rabbal Alamiin.

Buketrata, 12 Juli 2018 Penulis,

Said Mahfud NIM 1490343096

vi

# ABSTRAK

Perkembangan teknologi informasi saat ini sangat pesat, kebutuhan manusia dalam sumber data sangatlah penting. Untuk mengatasi perihal masalah tersebut penulis ingin membangun sistem *web server repositoy* berbasis *search engine*, yang merupakan layanan pustaka yang menyediakan beberapa sumber data dan informasi sebagai acuan untuk membuat suatu karya ilmiah. Pencarian pada sistem *repository* belum menggunakan algoritma pencarian sehingga hasil pencarian dari sistem tersebut belum optimal. Oleh karena itu, perlu adanya implementasi algoritma pencarian yang akan membantu menghasilkan hasil pencarian yang cepat dan optimal. Paper ini bertujuan untuk mengimplementasi Algoritma KMP pada fungsi pencarian dalam sistem *repository* pada prodi Teknologi Rekayasa Komputer Jaringan. Metode penelitian yang akan digunakan dalam penelitian ini yaitu tahap *System Development Life Cycle* yang terdiri atas analisis, rancangan sistem pencarian, implementasi algoritma KMP dan pengujian. Algoritma KMP berhasil diimplementasikan pada fungsi pencarian layanan sistem *repository journal* dan *e- book*. Hasil pengujian performa menampilkan bahwa rata-rata performa algoritma KMP dalam menemukan kata di form pencarian adalah 0.0116 detik. Hal ini membuktikan bahwa algoritma KMP sudah cukup cepat dan optimal dalam fungsi pencarian pada layanan sistem *repository journal* dan *e-book*.

Kata Kunci: *Web*, *Repository, algoritma KMP, server*

vii

**DAFTAR ISI DAFTAR GAMBAR DAFTAR TABEL**

viii

# BAB I PENDAHULUAN

* 1. **Latar Belakang**

Perkembangan teknologi khususnya dibidang teknologi informasi membuat semua pengguna membutuhkan teknologi yang cepat dan dapat diakses dari berbagai tempat dan berbagai sumber daya untuk mendukung segala jenis pekerjaan yang dilakukan sehari hari. Teknologi informasi telah bermetamorfosis menjadi sebuah basis penting dimana hal–hal substansial dari pengguna pribadi didokumentasikan dan disimpan dalam sebuah penyimpanan data. Dalam pelaksanaannya, penyimpanan data beserta aplikasi lainnya seringkali membutuhkan sumberdaya penyimpanan *Central Processing Unit* (CPU) yang statis atau tidak dapat dibawa kemana–mana, (Aisa, 2014).

Saat ini salah satu masalah yang sudah dapat diatasi adalah seperti masalah penyimpanan file. Proses menyimpan semua file baik foto, film, maupun tugas bahkan laporan pertanggung jawaban kegiatan selalu disimpan dalam media harddisk, flashdisk, ataupun penyimpanan yang lain. Semua orang yang ingin menyalin file pasti harus meminjam harddisk atau flashdisk tersebut lalu dipasangkan dari laptop yang satu ke laptop yang lain dan kembalikan ke pengguna. Hal tersebut tidak sesuai kaidah dan prinsip *smart technology for smart life*. Untuk itu, dengan adanya teknologi peneliti harus membuat suatu pemanfaatan teknologi baru dengan membuat media penyimpanan sendiri.

1

Pada Program studi Teknologi Rekayasa Komputer Jaringan saat ini masih belum menerapkan sistem *repository*. Berdasarkan hal tersebut, banyak mahasiswa dan dosen yang kesulitan dalam menyimpan dan mencari data informasi sebagai referensi untuk membuat suatu karya ilmiah. Untuk mengatasi masalah tersebut penulis akan membangun sebuah sistem *web server repository* yang merupakan layanan dari pustaka berupa pangkalan data dan informasi sebagai acuan untuk membuat suatu karya ilmiah. Pengguna dapat dengan mudah mendapatkan informasi yang terdapat pada *repository* dan dapat diakses melalui jaringan lokal maupun jaringan non lokal.

Untuk membangun sistem *repository* ini penulis membangunnya dengan menggunakan sistem *cloud storage* agar pengguna dapat mengakses dari jaringan non lokal. Pengguna dapat mengakses *repository* secara multiplatform baik dengan menggunakan pc, laptop atau *smartphone* yang mendukung jaringan internet.

# Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan, rumusan masalah yang diusulkan adalah sebagai berikut:

* + 1. Bagaimana merancang sistem untuk layanan *repository.*
    2. Bagaimana menerapkan algoritma *Knuth Morris Pratt* pada *search engine* untuk layanan *repository*.
    3. Bagaimana pencocokan *string* pada *pattern* secara manual.
    4. Bagaimana perhitungan kecepatan pencarian *string* untuk setiap percobaan pada sistem.

# Batasan Masalah

Pada tugas akhir ini terdapat pembatasan masalah dengan maksud untuk mempermudah penulisan. Adapun batasan masalah ini adalah sebagai berikut:

* + 1. Server data digunakan sebagai penyimpanan data serta informasi berupa jurnal dan karya ilmiah.
    2. Perangkat yang mengakses berupa PC, laptop dan *smartphone*.
    3. Jaringan LAN jurusan TIK.
    4. Studi kasus pada program studi Teknologi Rekayasa Komputer Jaringan.

# Tujuan Penelitian

Adapun tujuan yang dicapai dari penelitian ini antara lain adalah sebagai berikut:

* + 1. Melakukan Perancangan untuk sistem layanan repository.
    2. Menerapkan metode algoritma *Knuth Morris Pratt* pada pencarian sistem website.
    3. Melakukan perhitungan dan pecocokkan string secara manual pada pergeseran pattern.
    4. Mendata kecepatan perhitungan setiap pencarian kata pada sistem.

# Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat meningkatkan pengetahuan setiap pengguna layanan *Repository* tentang teknologi *cloud* sehingga meningkatkan produktifitas dan kinerja setiap pengguna. Manfaat yang diperoleh penulis adalah implementasi ilmu yang di dapat di perkuliahan serta pengetahuan baru yang tidak didapat dari perkuliahan.

# Road Map Penelitian

Penelitian dibidang teknologi *cloud*, terutama pada penyimpanan data *cloud computing*. Sebelumnya penelitian ini telah dilakukan oleh Rizky Syah Putra (2016). Dalam penelitian tersebut, ia menganalisa “Membangun Repository Politeknik Negeri Lhokseumawe Menggunakan Cloud Computing” hasilnya bertujuan untuk mempermudah mahasiswa dan dosen mencari data dan informasi serta mereka juga dapat menggunggah data dan informasi ke dalam sistem informasi.

Pada tahun 2017, Syahrizal telah melakukan sebuah penelitian dengan judul “Rancang Bangun Sistem Arsip Borang Akreditasi Pada Program Studi Teknik Informatika Dengan Metode *Searching Knuth Morris Pratt*”, hasil dari pengujian sistem untuk mengetahui sejauh mana tingkat keberhasilan sistem dalam memberikan informasi kepada user serta mengetahui kelebihan dan kekurangan sistem ini.

Aisa telah melakukan sebuah penelitian dengan judul Implementasi *Private Cloud* Menggunakan *Raspberry Pi* untuk pengaksesan data pribadi. Hasil

penelitian: sistem *Private Cloud Computing* menggunakan *Raspbery Pi* dinilai sangat bermanfaat dan berguna bagi pengguna teknologi. Hal ini disebabkan sistem *private cloud computing,* yang berupa sistem atau sumber daya yang sangat kecil, dapat diakses secara *online*. Tujuan penelitian ini mengimplementasikan *private cloud* menggunakan *Raspberry Pi,* sehingga dapat memberikan kemudahan bagi pengguna pribadi untuk mengakses data dari mana saja dan dapat menjalankan aplikasi basis data sederhana dari *Raspberry Pi* yang dijadikan *server* mini (Aisa, 2014).

Implementasi *Algoritma Knuth-Morris-Pratt* Pada Fungsi Pencarian Judul Tugas Akhir *Repository*. Hasil penelitian : untuk melakukan penerapan algoritma pencaharian yang akan membantu menghasilkan hasil pencaharian yang cepat dan optimal. Pada hasil pengujian peforma menunjukan bahwa rata-rata performa algoritma KMP dalam menemukan kata di form pencaharian adalah 0.0138 detik. Hal ini menunjukan bahwa algoritma KMP sudah cukup cepat dan optimal dalam fungsi pencarian sebuah repository jurusan, (sa'diah, 2017).

Pasang Surut Pengelolaan *Institutional Repository (IR)* Perpustakaan Universitas Muhammadiyah Surakarta. Hasil Penelitian: membahas perkembangan Instusi *repository* yang terjadi di Pepustakaan Universitas Muhammadiyah Surakarta. Beberapa dasar pemikiran IR merupakan sumber infromasi yang terbuka dan mengurangi plagiasi, mencegah duplikasi penelitian, sebagai media promosi dan akhirnya IR mampu memberikan kontribusi mengangkat citra dan reputasi penguruan tinggi dalam ranah webometrik (Widada, 2016).

Membangun *Web Server Repository* Ubuntu 10.04 di Laboratorium Sinyal dan Sistem Politeknik Negeri Lhokseumawe. Hasil penelitian: Untuk menguji kemampuan *repository* server memungkinkan user melakukan automatisasi proses instalasi, upgrade dan konfigurasi paket software dari komputer. Kecilnya bandwidth internet menyebabkan kesulitan instalasi paket Ubuntu Linux dari server *repository* karena proses instalasi sangat lambat. Dengan adanya *Local repository server* di Laboratorium Sinyal dan sistem di Politeknik Negeri Lhokseumawe, maka mahasiswa pengguna linux Ubuntu khususnya dapat mengatasi kesulitan instalasi paket Ubuntu pada penambahan aplikasi-aplikasi yang diinginkan dari *local repository* server dengan mudah dan lebih cepat (Mulyadi, 2008).

# Sistematika Penulisan

Untuk memudahkan pemahaman terhadap Tugas Akhir ini maka penulis menyusun sistematika penulisan sebagai berikut

# BAB I PENDAHULUAN

Bab ini merupakan pendahuluan yang berisikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, road map, metode penulisan dan sistematika penulisan.

# BAB II TINJAUAN PUSTAKA

Bab ini membahas tentang teori dasar yang mendukung dan mendasari dengan pembangunan layanan *repository* berbasis *search engine.*

# BAB III METODE PENELITIAN

Bab ini membahas tentang Perancangan sistem, Diagram penelitian, Perancangan DFD dan ERD, Flowchart, Implementasi sistem dan perancangan *user interface*.

# BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang hasil dan pembahasan tentang program yang terdiri dari penjelasan penelitian.

# BAB V PENUTUP

Bab ini merupakan bagian penutup yang berisi simpulan dan saran yang didapatkan dari hasil implementasi tugas akhir ini.

# BAB II TINJAUAN PUSTAKA

* 1. **Pemanfaatan Media Penyimpanan *Cloud***

Pemanfaatan Media *cloud* sebagai media penyimpanan data dengan jaringan internet dalam bentuk dokumen file jurnal dan *e-book* sangat membantu untuk memajukan sebuah instansi atau kampus dan memberikan hak akses kepada setiap kalangan masyarakat untuk dapat melihat dan membacanya. Banyak perusahaan atau individu menyimpan data penting mereka dalam *cloud*. Menurut mereka, menyimpan data di internet menjadi alternatif utama daripada harus menyimpan dalam bentuk fisik karena kecenderungan kerusakannya menurut mereka lebih tinggi.

*Cloud* merupakan sebuah kemajuan dan perkembangan teknologi dimana dapat mempermudah manusia yang bertindak sebagai user dan merupakan sebuah sarana untuk menyimpan data pribadi dengan kapasitas tertentu. Untuk setiap metode pengaksesannya membutuhkan koneksi internet, kapanpun dan dimanapun selama terdapat koneksi internet yang stabil maka *cloud* dapat dijangkau. Disini dapat dilihat bahwa *cloud* bersifat fleksibel, tetapi dengan kelebihan yang dimiliki oleh layanan *cloud* ini, terdapat juga beberapa risiko yang dapat terjadi dan tidak bisa dihindari jika telah berada pada jaringan internet bersifat *public*, memperkuat sistem keamanan adalah kunci utama agar data pengguna tetap terjaga.

*Experimental model of Low Power NAS and Cloud Drive Based on Raspberry Pi*. Hasil penelitian: Untuk Teknologi mobile dan khususnya

8

penggunaan *cloud space* berhasil terintegrasi di web dan penggunanya semakin meningkat setiap harinya. Ide *cloud* adalah setiap pengguna memiliki tempat sendiri untuk menyimpan informasi yang bisa diakses dari mana saja. Sebagian besar biaya layanan tergantung pada jumlah ruang yang digunakan. Idenya adalah menggunakan *Raspberry Pi* sebagai jaringan penyimpanan, yang nantinya bisa dijadikan *cloud server* (Kandov, 2015)**.**

Sistem Repositori Tugas akhir Mahasiwa dengan Fungsi Peringkat Okapi BM25. Hasil penelitian : untuk melakukan pencarian dibuat dalam bentuk query sederhana dengan kriteria yang masih terbatas, tanpa pengurutan dengan peringkat. Selain itu, kepala lab di jurusan juga menemui kesulitan dalam melakukan pemetaan bidang keahlian dari tugas akhir yang dikerjakan oleh mahasiswa di masing-masing lab. Pencarian tugas akhir pada penelitian ini berdasarkan query yang diinput oleh pengguna menggunakan metode pencarian fungsi Okapi BM25. Dengan fungsi peringkat Okapi BM25 maka hasil karya dapat ditampilkan dengan urutan peringkat sesuai relevansinya (Widiasri, 2016).

## *Cloud Storage*

[*Cloud Storage*](http://cloudindonesia.blogdetik.com/mengenal-cloud-storage)adalah media penyimpanan *online*, di mana Anda dapat menyimpan data pada server virtual yang tersedia. Dengan adanya *Cloud Storage*, Anda tidak perlu lagi menyimpan data Anda pada *hard drive*, CD, ataupun *hardware* lainnya. Anda tinggal menyimpan data Anda pada *remote database* yang disediakan oleh pihak ketiga, untuk mengaksesnya Anda hanya memerlukan

koneksi internet. Menggunakan [*cloud storage*](http://cloudindonesia.com/mengenal-cloud-storage-media-penyimpanan-online)sebagai media penyimpanan data akan memberikan Anda beberapa keuntungan, di antaranya:

1. Anda dapat mengakses data Anda yang tersimpan pada *Cloud Storage* dari manapun dan kapanpun, selama ada koneksi internet.
2. Anda tidak perlu membawa banyak CD atau membawa *hard disk eksternal*, karena data Anda tersimpan pada *database virtual*.
3. Jika diperlukan, Anda dapat berbagi data Anda yang tersimpan pada [*Cloud Storage*](http://cloudindonesia.com/mengenal-cloud-storage-media-penyimpanan-online) dengan mudah.

Namun, disamping keuntungan yang bisa didapatkan, *Cloud Storage* juga memiliki beberapa kelemahan. Kelemahan yang paling utama terletak pada hal keamanan. Untuk menyimpan data Anda pada *Cloud Storage* ini tentunya Anda diharuskan membuat sebuah akun. Setiap akun dilindungi oleh password yang bisa saja diketahui orang lain jika Anda tidak berhati-hati, sehingga besar kemungkinan akun Anda dapat dibuka oleh orang lain. Kelemahan lain adalah gangguan yang terjadi pada saat mengakses data Anda pada *Cloud Storage*, hal ini dapat disebabkan oleh koneksi internet yang bermasalah atau server yang sedang *down*. Yang terakhir, untuk mengakses data Anda pada *Cloud Storage* diperlukan koneksi internet, sehingga tanpa koneksi internet Anda tidak dapat mengakses data anda. Sampai saat ini terdapat beberapa *provider* yang menyediakan layanan [*Cloud*](http://cloudindonesia.com/mengenal-cloud-storage-media-penyimpanan-online)[*Storage*](http://cloudindonesia.com/mengenal-cloud-storage-media-penyimpanan-online) yang mungkin familiar di telinga anda, namun Anda tidak menyadari kalau layanan yang mereka berikan itu merupakan layanan *Cloud Storage*. Beberapa diantaranya adalah Evernote, Google Docs, Flickr dan Picasa, RapidShare, 4Shared, dan masih banyak lagi.

* 1. **Layanan *Cloud Storage***

Berdasarkan jenis layanannya *cloud storage* dibagi menjadi 3 (tiga) macam,

yaitu:

* + 1. *Software as a Service (SaaS)*: Layanan yang menyediakan aplikasi jadi / siap pakai kepada *End user*. Ciri dari layanan ini adalah user tidak perlu membuat aplikasi, tidak perlu menyiapkan tempat dan juga infrastruktur. Contoh SaaS adalah gmail, ymail, facebook, twitter, dropbox. Atau yang berbayar seperti salesforce, office365, dsb.
    2. *Platform as a Service (PaaS)*: Layanan yang menyewakan “tempat” untuk menjalankan aplikasi dari user. Tempat yang dimaksud seperti sistem operasi, database, framework, dsb yang merupakan wadah untuk berjalannya aplikasi. Ciri dari layanan ini adalah user tidak perlu melakukan maintenance dan tidak perlu menyiapkan infrastruktur. Sehingga user dapat tetap fokus membangun aplikasinya. Contoh Pass adalah Windows Azure, Amazon Web Service, GoogleApp Engine.
    3. *Infrastructure as a Service (IaaS)*: Layanan yang menyewakan infrastruktur IT kepada user yang ingin membangun layanan cloud. Infrastruktur disini bersifat fisik, bisa berupa memory, penyimpanan, server, jaringan, dsb. Hal- hal seperti membuat aplikasi dan konfigurasinya diserahkan kepada user. *Cloud provider* hanya menyediakan infrastruktur berdasarkan request dari user. Ciri layanan ini adalah jika user ingin mengupgrade *memory* atau

menambah server, user tinggal menghubungi *provider* kemudian *provider* akan menyediakan sesuai dengan permintaan. Contoh IaaS adalah Amazon EC2, *Rackspace cloud*.

* 1. **Algoritma *Knuth Morris Pratt***

Algoritma KMP dikembangkan oleh D.E. Knuth, J.H.Morris, dan V.R. Pratt. Algoritma ini merupakan jenis *extract string matching algorithm* yang melakukan pencocokan string secara tepat sesuai dengan susunan karakter. Algoritma KMP merupakan algoritma pencarian hasil pengembangan dari algoritma pencarian Brute Force. Algoritma ini memiliki keunggulan pencarian kecocokan pada file yang berukuran besar. Algoritma KMP mencari teks berdasarkan urutan dari kiri ke kanan pada awal teks dan kemudian menggeser susunan kata sampai pada ujung teks. Penelitian mengenai implementasi algoritma algoritma KMP sudah pernah dilakukan oleh Rossaria et al (2015), yaitu implementasi algoritma KMP dalam aplikasi dokumen berbasis android. Selain itu, Zaky (2015) mengimplementasikan algoritma KMP dalam perancangan game HANACARAKA.

Algoritma *Knuth-Morris-Pratt* merupakan proses pencocokan string. Bila terjadi ketidak cocokan pada saat *pattern* sejajar dengan teks [i..i + n -1], kita bisa menganggap ketidak cocokan pertama terjadi di antara teks [i + j] dan pattern [j], dengan <j<n. Berarti, teks [i..i + j]=pattern [0..j + 1] dan a=teks [i+j] tidak sama dengan b=pattersn [j], ketika kita menggeser (Riyanarto Sarno, Yeni Anistyasari, dan Rahimi Fitri, 2012 : 18).

* + 1. **Tahapan kerja Algoritma *Knuth Morris Pratt***

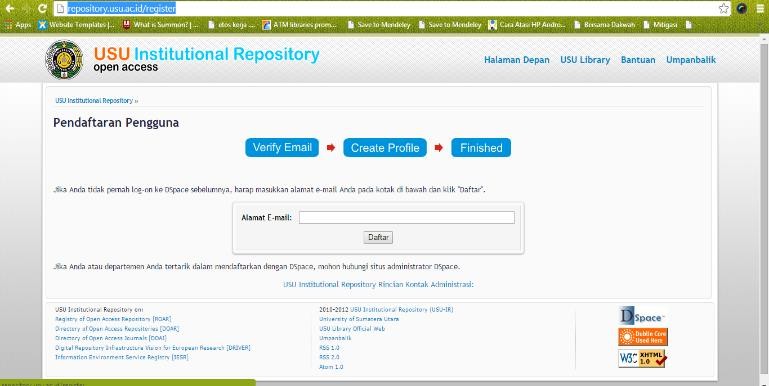
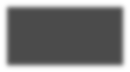
Pencocokan string akan berjalan secara efesien bila kita mempunyai tabel yang menentukan berapa panjang kita seharusnya menggeser seandainya terdetek ketidakcocokan di karakter ke-j dari *pattern*. Tabel itu harus memuat *next*[j] yang merupakan posisi karakter *pattern*[j] setelah digeser, sehingga kita menggeser *pattern* secara besar j-*next*[j] relatif terhadap teks. Secara sistematis, langkah- langkah yang dilakukan algoritma *Knuth-Morris-Pratt* pada saat mencocokan string adalah sebagai berikut:

* + - 1. Algoritma *Knuth-Morris-Pratt* mulai mencocokan *pattern* pada awal teks.
      2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter pattern, dengan karakter di teks yang bersesuaian sampai salah satu kondisi berikut terpenuhi:
         1. Karakter di pattern dan diteks yang dibandingkan tidak cocok (*mismatch*).
         2. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan diposisi ini.

Algoritma kemudian menggeser *pattern* berdasarkan *table next*, lalu menghitung langkah 2 sampai *pattern* berada di unjung teks (Waruwu, 2016).

* 1. **Definisi *Repository***

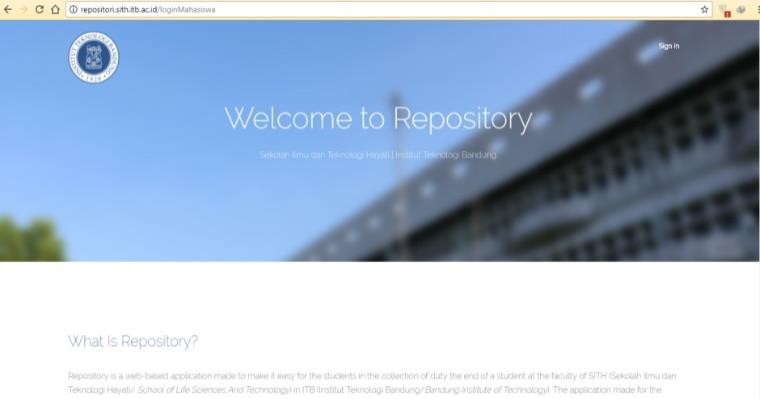
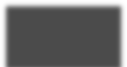
Menurut Jonner Hasuglan, secara sederhana arti dari repositori adalah tempat penyimpanan. Dalam konteks kepustakawanan repositori adalah suatu tempat dimana dokumen, informasi atau data disimpan, dipelihara dan digunakan. Kadang-kadang istilah *repository* dipakai untuk menyatakan hal yang sama.



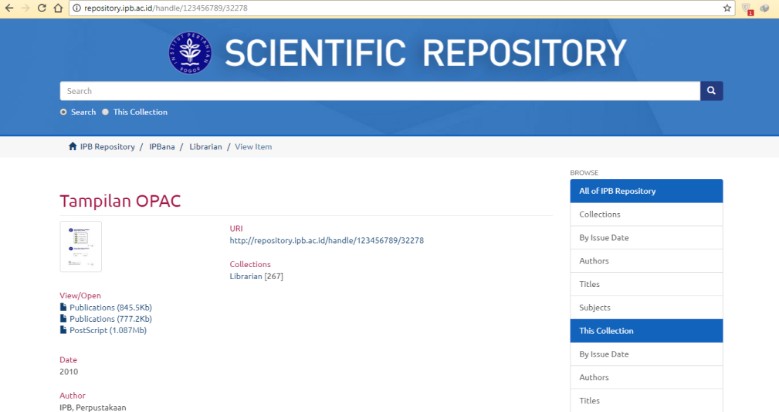
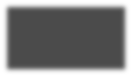
Gambar 2.1 Tampilan Repositori USU

(Kanbudhir, 2016)

*Repository* adalah tempat di simpannya berbagai macam program atau aplikasi yang telah di buat sedemikian rupa sehigga bisa di akses melalui internet. Selain di internet, *repository* juga tersedia di media seperti DVD sebagai alternatif *repository* saat tak ada koneksi internet. Ketika kita melakukan *download repository* melalui internet, default servernya adalah server luar seperti archive.ubuntu.com, security.ubuntu.com, dll. Untuk fungsinya, disini *repository* berperan sebagai penyedia aplikasi atau kumpulan paket software dari distro-distro linux, yang dapat di akses melalui internet.



Gambar 2.2 Tampilan Repositori ITB (Ganesha, 2017)



Gambar 2.3 Tampilan Repositori IPB

(Rosdiana, 2015)

Penguruan tinggi melakukan aktifitas mengumpulkan, melestarikan dan menyebarluaskan output berupa karya ilmiah dan jurnal dalam format digital secara online maupun *offline*. Hal ini dilakukan secara berkesinambungan untuk menunjang kegiatan akademik dan penelitian. Ada 4 aspek yang menjadi perhatian utama bagi eksistensi institusional *repository* sebuah perguruan tinggi, diantaranya sebagai berikut:

* + 1. Untuk mengumpulkan konten dalam satu lokasi sehingga mudah untuk ditemukan kembali.
    2. Untuk menyimpan dan melestarikan aset intelektual sepanjang waktu.
    3. Untuk menyediakan akses terbuka terhadap karya intelektual institusi kepada khalayak umum.
    4. Untuk menciptakan visibilitas global bagi karya ilmiah institusi.
  1. **Konsep Pengembangan *Repository***

Langkah pembangunan serta pengembangan sebuah institusional repository wajib dilandasi dengan adanya rencana pengembangan jangka panjang yang jelas dan strategis. Strategi pengembangan sistem informasi terbagi atas 3 konsepsi yakni:

* + 1. Perlunya membuat kesamaan pandangan tentang pentingnya sistem informasi.
    2. Rencana aksi secara gamblang dilakukan seluruh unit.
    3. Seluruh aspek bisnis sudah ditopang oleh sistim informasi.

Dengan penerapan 3 konsepsi tersebut diharapkan sistem informasi sudah secara otomatis digunakan untuk proses pengambilan keputusan. Dengan demikian fakta dilapangan yang sering kita alami yakni subjektifitas dalam pengambilan keputusan manusia yang berbasiskan perasaan dapat kita hindari (Putra, 2016).

* 1. **Debian *Server***

Debian adalah sistem operasi bebas yang dikembangkan secara terbuka oleh banyak programer sukarela (pengembang Debian) yang tergabung dalam Proyek Debian. Sistem operasi Debian adalah gabungan dari perangkat lunak yang dikembangkan dengan lisensi GNU, dan utamanya menggunakan kernel Linux, sehingga populer dengan nama Debian GNU/Linux. Sistem operasi Debian yang menggunakan kernel Linux merupakan salah satu distro Linux yang populer dengan kestabilannya.

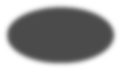
Dengan memperhitungkan distro berbasis Debian, seperti Ubuntu, Xubuntu, Knoppix, Mint, dan sebagainya, maka Debian merupakan distro Linux yang paling banyak digunakan di dunia. Program kanonik untuk menangani paket-paket tersebut adalah dpkg, paling sering melalui apt / aptitude. deb paket dapat dikonversi ke paket lain dan sebaliknya menggunakan asing. Beberapa paket Debian inti tersedia sebagai udebs (“mikro deb”), dan biasanya hanya digunakan untuk bootstrap instalasi Linux Debian. Meskipun file tersebut menggunakan ekstensi nama file udeb, mereka mematuhi spesifikasi struktur yang sama seperti biasa deb. Namun, tidak seperti rekan-rekan mereka, deb hanya berisi paket-paket udeb fungsional penting file. Secara khusus, file dokumentasi biasanya dihilangkan. Udeb paket tidak dapat diinstal pada sistem Debian standar, (Maysaroh, 2012).



Gambar 2.4 Logo Debian Linux (Pedro pinto, 2017)

## PHP *(Hypertext Preprocessor)*

PHP adalah bahasa pemrograman berbasis *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh *The PHP Group*. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client).*



Gambar 2.5 Bahasa PHP

(Sumber: [www.php.net](http://www.php.net/))

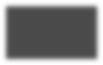
Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat *website* pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman *web* yang *powerful* dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti *wikipedia*, *wordpress*, *joomla*, dll.

Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri PHP: *Hypertext Preprocessor*. PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. PHP dirilis dalam lisensi PHP *License*, sedikit berbeda dengan lisensi GNU *General Public License* (GPL) yang biasa digunakan untuk proyek *Open Source,* (Andre, 2014).

## *Web Server*

Web server merupakan perangkat lunak yang diletakkan didalam komputer server yang bersifat pasif, yang tugasnya adalah untuk melayani permintaan (*request*) dari komputer klien yang bersifat aktif didalam jaringan komputer dengan menggunakan protocol *hypertext transfer protocol* (HTTP) atau *hypertext Transfer Protocol secure* (HTTPS). Web server Apache dipilih untuk digunakan karena mudah dalam konfigurasi, mendukung untuk ditananamkan dalam modul *Raspberry Pi* dan dapat digunakan secara gratis. Menurut Akbar iin, Web server merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan

mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Software Web server diantaranya: Apache, IIS, Xitami, dan Zeus web Server. Namun web yang terkenal dan yang sering digunakan adalah Apache dan Microsoft Internet Information Service (IIS).



Gambar 2.6 Webserver Apache (Sumber: [www.](http://www/) news-coffee.com)

## *Xampp*

Software yang satu ini bernama XAMPP yang merupakan singkatan dari Apache, MySQL, PHP dan Perl sedangkan huruf “X” dimaksudkan sebagai suatu software yang dapat dijalankan di [empat OS](https://www.nesabamedia.com/pengertian-dan-fungsi-sistem-operasi/) utama seperti Windows, Mac OS, Linux dan Solaris. Istilah ini seringkali disebut dengan *cross platform* (*software multi OS*). Sesuai dengan namanya software yang satu ini merupakan gabungan dari beberapa software dengan fungsi yang sama yakni menunjang para pembuat web yang menginginkan adanya web server sendiri di PC atau laptopnya. Software ini juga berlisensi GNU dan dapat didownload secara gratis di internet mengingat peran vital yang dimilikinya terutama bagi pembuat web pemula.

*Software* XAMPP didirikan oleh suatu perusahaan bernama *Apache Friends*. Dengan adanya beberapa tools pemrograman seperti MySQL, PHP dan Perl yang dimilikinya tentu mengindikasikan jika anda menekuni salah satu atau semuanya berarti harus memiliki software yang bernama XAMPP ini. Maksud dari Apache yakni selain mengindikasikan nama pengembangnya juga merupakan suatu software yang menghadirkan [*web server*](https://www.nesabamedia.com/pengertian-web-server/) pada komputer anda layaknya web server sesungguhnya (Firmansyah, 2017).



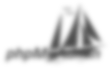
Gambar 2.8 Software Xampp (Firmansyah, 2017)

## *Database MySQL*

*Database* adalah sebuah koleksi dokumen yang terstruktur atau data yang tersimpan pada sebuah sistem komputer dan diatur sedemikian rupa sehingga dapat dicari dengan cepat dan informasi bisa segera didapatkan. SQL pada MySQL adalah akronim dari *Structured Query Language*. Sebuah database MySQL berisi satu atau banyak table, setiap table berisi *record* (data) atau *row* (baris). Dalam baris ini terdapat berbagai *column* (kolom) atau *fields* yang berisi data.

Phpmyadmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, anda dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate

data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual. PHPMyAdmin merupakan *front-end* MySQL berbasis web. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia *hosting* yang ada di internet. PHPMyAdmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengeksport data ke dalam berbagai format data. PHPMyAdmin juga tersedia dalam 50 bahasa lebih, termasuk bahasa Indonesia.



Gambar 2.9 Logo phpMyAdmin (Sumber: [www.phpmyadmin.net](http://www.phpmyadmin.net/))

*Database* (basis data) dapat diungkapkan sebagai suatu pengoperasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah dan menghapus data, (alamsyah, 2015).

## *PuTTY*

*PuTTY* adalah sebuah program *open source* yang dapat Anda gunakan untuk melakukan protokol jaringan SSH, Telnet dan Rlogin. Protokol ini dapat digunakan untuk menjalankan sesi remote pada sebuah komputer melalui sebuah jaringan, baik itu LAN, maupun internet. Program ini banyak digunakan oleh para pengguna komputer tingkat menengah ke atas, yang biasanya digunakan untuk menyambungkan, mensimulasi, atau mencoba berbagai hal yang terkait dengan jaringan. Program ini juga dapat Anda gunakan sebagai tunnel di suatu jaringan.

# BAB III METODELOGI PENELITIAN

* 1. **Identifikasi Masalah**

Masalah yang timbul di ruang lingkup Program Studi Teknologi Rekayasa Komputer Jaringan adalah proses mengungggah jurnal dan karya ilmiah untuk kebutuhan referensi mahasiswa yang ingin mempelajari sistem dan penelitian. Pada zaman sekarang ini dimana teknologi telah mempengaruhi semua kalangan dan pekerjaan maka memanfaatkan media informasi yang canggih untuk dapat mempermudah dosen maupun mahasiswa dalam berbagi file informasi yang dapat dilakukan kapanpun dan dimanapun.

# a. Analisis Sistem

Terdapat 3 menganalisis sistem untuk Layanan Repository dengan menerapkan metode algoritma pencarian berupa Knuth Morris Pratt dalam proses pencarian judul jurnal ataupun karya ilmiah. Berikut merupakan analisis sistem:

1. Analisis Deskriptif

Dalam analisis deskriptif ini memberikan gambarkan pengaruh Layanan Repository dengan menerapkan metode algoritma pencarian berupa Knuth Morris Pratt dalam proses pencarian judul jurnal ataupun karya ilmiah, dimulai dari rancang bangun website, pengujian system sampai hasil.

24

1. Analisis Komperatif

Dalam analisis komperatif ini melakukan pengujian Layanan Repository yang diakses oleh member dan semua kalangan dalam jaringan yang WLAN (Wireless Local Area Network).

1. Analisis Kebutuhan

Analisis kebutuhan bagian dari spesifikasi peralatan yang digunakan dalam pengaruh Layanan Repository dengan menerapkan metode algoritma pencarian berupa Knuth Morris Pratt dalam proses pencarian judul jurnal ataupun karya ilmiah, spesifikasi peralatan yang digunakan dari pembuatan sampai dengan pengujian serta hasil adalah sebagai berikut :

* 1. Perangkat Keras (Hardware)
     1. Satu unit computer dengan hardisk 500GB
     2. Satu unit laptop ASUSTek Computer S451LB dengan
     3. Processor INTEL inside core i3,
     4. Memory 4GB DDR3,
     5. Dan harddisk 500 GB.
  2. Perangkat Lunak (Software)

1. Website Layanan Repository yang dirancang dengan menggunakan bahasa pemograman antara lain:
   1. Php MySQL
   2. Java Script
   3. CSS
   4. HTML
   5. Bootstrap
2. Sistem Operasi Windows 10
3. Microsoft Office Visio 2016 untuk merancang pemodelan sistem.
4. Sublime sebagai aplikasi untuk text editor program.
5. Sistem Operasi Debian 9 GUI untuk membangun server local.
6. Adobe Photoshop untuk merancang interface web server.

# Analisa Algoritma

Dalam proses perancangan dan implementasi dari Layanan Repository ini berbasis search engine dengan algoritma penerapan Knuth Morris Pratt. Pembelajaran Knuth Morris Pratt yang mencocokan kumpulan string yang menggabungkan berbagai proses pencarian dalam suatu Layanan Repository pada program studi Teknologi Rekayasa Komputer Jaringan sangatlah penting, namun terkadang proses pencarian itu lambat. Untuk mempercepat dan mempermudah suatu proses pencarian, dibutuhkan suatu algoritma ini. Dengan penerapan Algoritma KMP akan mempermudah pengunjung untuk dapat mencari jurnal atau karya ilmiah sesuai dengan kata kunci yang dimasukkan.

# Perancangan Sistem

* + 1. **Perancangan *Web Application***

Perancangan Web Layanan *Repository* berbasis *Search Engine*

menggunakan algoritma *Knuth Morris Pratt* memerlukan perancangan yang baik.

Dalam perancangan menyangkut semua elemen yang membentuk *Web Application*. Adapun langkah-langkah perancangan *Web Application* sebagai berikut :

1. *Use Case* Diagram untuk perancangan sistem

Berikut ini merupakan operasional sistem yang penulis uraikan dengan *use diagram* pada gambar 3.2 :



**Sistem Repository**

Registrasi

Login

Mengolah Data Member

Mempublikasi Jurnal

MEMBER

Mengunggah Jurnal

ADMIN

Mengunduh Jurnal

Mengubah Data Pribadi

Memverifikasi Data Member

VISITOR

Mengolah Data Jurnal

Logout

Gambar 3.2 *Use Case* Diagram Sistem *Repository*

Pada gambar 3.2 menunjukkan pelaku sistem dari *repository* pada prodi Teknologi Rekayasa Komputer Jaringan menggunakan server *Debian Linux*, sekaligus mendeskripsikan aliran data apa saja yang dibutuhkan entitas dalam tiap-

tiap aktivitas yang dilakukan, berikut penjelasan singat untuk mudah memahaminya.

* 1. Admin mengelola data login user, menverifikasi setiap user, mengelola data jurnal dan menyetujui setiap jurnal yang masuk agar bisa di publikasi ke menu utama, admin dapat mengetahui data jurnal yang diupload oleh member dan juga dapat di download.
  2. Member dapat Mengupload jurnal dan melihat daftar jurnal yang telah diupload, dan jurnal yang akan diupload memiliki dua tipe: *publish* dan *not publish*. Jurnal yang *publish* akan dapat dilihat oleh semua pengunjung web sedangkan jurnal yang bersifat *not publish* hanya dapat dilihat oleh member yang mengupload saja. Member juga dapat mengubah data pribadi sesuai keinginan.
  3. Visitor dapat mengakses halaman utama pada layanan system

*Repository* dan juga dapat mengunduh file sesuai kebutuhan.

1. Diagram Alur Sistem

Merupakan skema jalannya sistem layanan sistem Repository pada program studi Teknologi Rekayasa Komputer Jaringan, untuk lebih jelasnya dapat melihat gambar 3.3 dibawah ini.

**YA**

Mengunggah File

**Tidak**

Login User

Tampilan Awal repository

**Tidak**

Mengunduh File

Tampilan Jurnal dan TA berdasarkan pencarian

Mencari Jurnal dan TA

MULAI

Tampilan Jurnal yang sudah diupload

END

Langsung Unduh

Memilih Jurnal dan TA

**YA**

Gambar 3.3 Diagram Alur Sistem *Repository*

# Perancangan Database

Adapun dalam perancangan database yang harus dirancang adalah DFD (*Data Flow Diagram*), ERD (*Entity Relationship Diagram*) yang digunakan untuk pembuatan sistem database seperti berikut:

1. Perancangan Diagram Konteks pembuatan Web Repository

Diagram Konteks digunakan untuk menunjukkan entitas-entitas yang menunjukkan pelaku sistem dari web Layanan Repository menggunakan algoritma searching pada Program Studi Teknologi Rekayasa Komputer Jaringan, serta menggambarkan aliran data apa saja yang dibutuhkan , entitas dalam tiap-tiap aktifitas yang dilakukan dan agar lebih mudah dipahami

1. Pembuatan *Entity Relationship Diagram*

Dalam pembentukan ERD terdapat 3 komponen yang dapat dibentuk, yaitu entitas, relasi, dan atribut. Entity Relationship Diagram (ERD) pada sistem layanan repository prodi. Berikut merupakan ERD Web Repository yang terdapat pada gambar 3.5:

Menginput

Member

Admin

Id\_jur

nim

Text\_message

status

markDate

Id\_mes

N

Message

nim

N

username

name

Membuat

Mendapat

alamat

Jenis\_kelamin

id

1

1

Hak\_akses

username

status

Email\_Adm

jurusan

password

semester

Status\_adm

1

1

email

No\_hp

Mendapat

N

N

Membuat

password

regdate

N

Id\_jurnal

status

Nama\_file

Mendapat

file

Tanggal\_upload

nim

abstrak

Jenis\_file

1

Ukuran\_file

Download

Id\_jur

keyword

deskripsi

judul

url

Jurnal

Gambar 3.4 *Entity Relationship Diagram*

1. Tabel Admin

Tabel admin digunakan untuk menyimpan data admin yang bersifat pribadi yang terdiri dari beberapa field seperti pada tabel 3.1:

Tabel 3.1 Tabel Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Keterangan** |
| Id | Int (2) | Primary key | ID Admin |
| Username | Varchar (30) |  | Username Admin |
| Email\_adm | Varchar (30) |  | E-mail Admin |
| Password | Varchar (20) |  | Password Admin |
| Status\_adm | Int (2) |  | Status Admin |

1. Tabel Member

Tabel member digunakan untuk menyimpan semua data mahasiswa yang terdiri dari beberapa field seperti pada table 3.2:

Tabel 3.2 Tabel Member

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Keterangan** |
| Nim | Int (12) | Primary key | NIM Member |
| Username | Varchar (15) |  | Username Member |
| Nama | Varchar (30) |  | Nama Lengkap Member |
| Alamat | Varchar (30) |  | Alamat Member |
| Jenis\_kelamin | Varchar (15) |  | Jenis Kelamin Member |
| Hak\_akses | Varchar (10) |  | Izin Akses Member |
| Status | Varchar (12) |  | Status Member |
| Jurusan | Varchar (20) |  | Jurusan Member |
| Semester | Varchar (10) |  | Semester Member |
| Email | Varchar (30) |  | E-mail Member |
| No\_hp | Varchar (15) |  | No Telp Member |
| Password | Varchar (10) |  | Password Member |
| regDate | Timestamp |  | Tanggal Register Member |

1. Tabel Message

Tabel Message digunakan untuk mengirimkan pesan konfirmasi kepada member terkait file jurnal yang telah di setujui maupun tidak oleh admin, yang terdiri dari beberapa field seperti pada table 3.3 berikut:

Tabel 3.3. Tabel Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Keterangan** |
| Id\_mes | Int (11) | Primary key | ID Pesan |
| Id\_jurnal | Int (11) | Foreign key | ID Jurnal |
| Nim | Int (12) | Foreign key | NIM Member |
| Status | Varchar (20) |  | Status Pesan |
| Text\_message | Mediumtext |  | Isi Pesan |
| markDate | Timestamp |  | Tanggal Masuk Pesan |

1. Tabel Download

Table download digunakan untuk menyimpan data hasil upload jurnal dari masing- masing member yang telah mengunngah file pada dashboard upload, yang

terdiri dari beberapa field seperti pada table 3.4 berikut:

Tabel 3.4 Tabel Download

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Keterangan** |
| Id\_jurnal | Int (11) | Primary key | ID Jurnal |
| Tanggal\_upload | Date |  | Tanggal Unggah Jurnal |
| Nama\_file | Varchar (30) |  | Nama Jurnal |
| Abstrak | text |  | Abstrak Jurnal |
| Jenis\_file | Varchar (20) |  | Jenis File Jurnal |
| Ukuran\_file | Varchar (20) |  | Ukuran File Jurnal |
| File | Varchar (20) |  | Folder Tujuan Jurnal |
| Nim | Int (12) | Foreign key | NIM Member |
| Status | Varchar (20) |  | Status Jurnal |

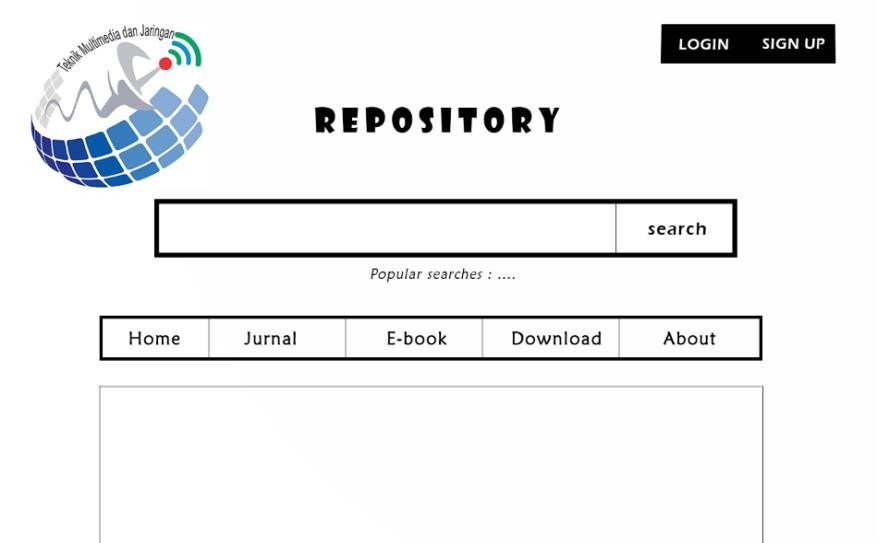
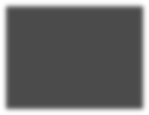
# Rancangan Desain Interface Web

Adapun Rancang desain interface dalam perancangan sistem Web layanan

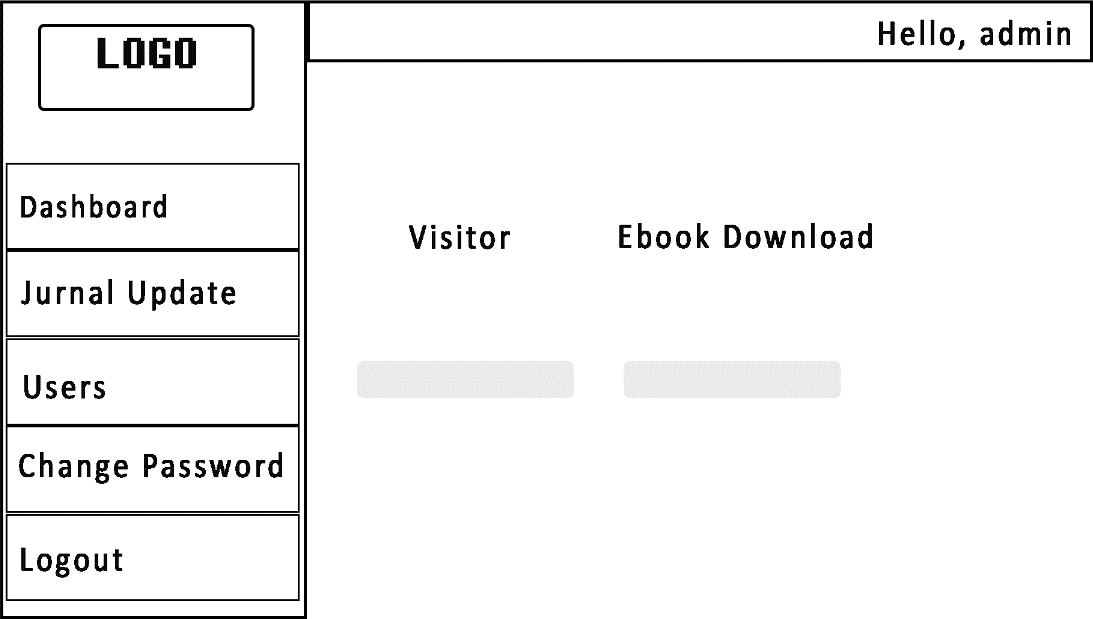
*Repository* sebagai berikut :

1. Tampilan Homepage

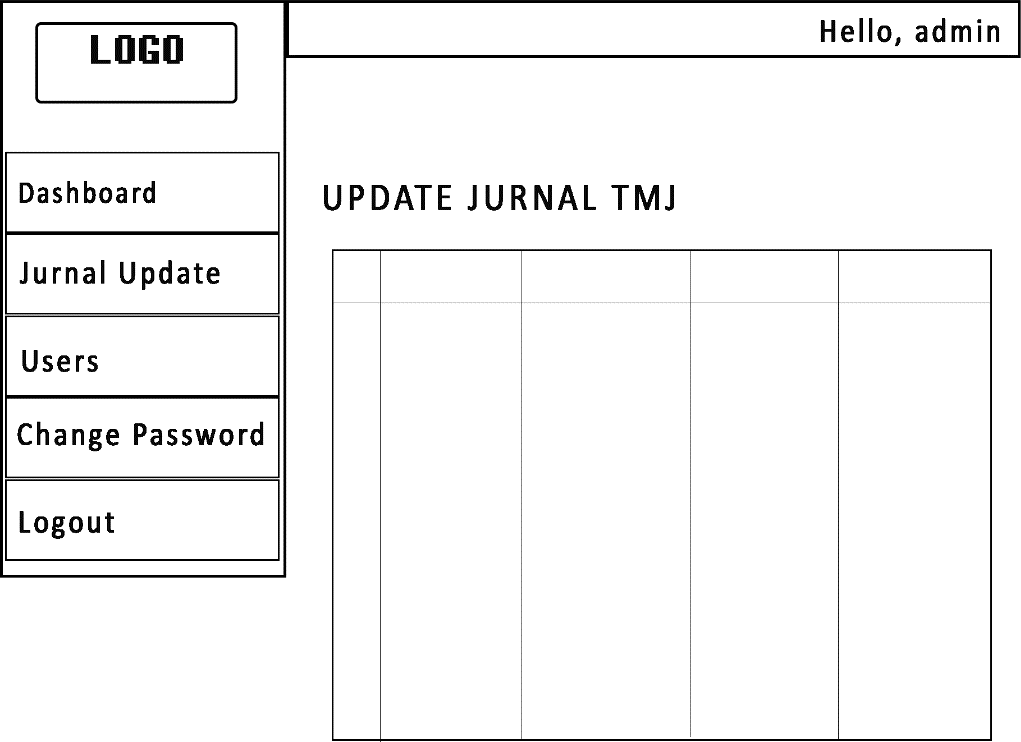
Pada gambar 3.6 merupakan tampilan halaman utama web yang dimaksud juga dengan desain interface keseluruhan sebelum user masuk ke menu yang diinginkan. Pada halaman ini , Web layanan *Repository* menyediakan tampilan seperti menu Home, , Searchbar, Login, Tentang visi misi program studi yang berisikan informasi Web, Login yang dapat diakses Oleh Admin dan member dimana login ini bersifat Multilevel user.



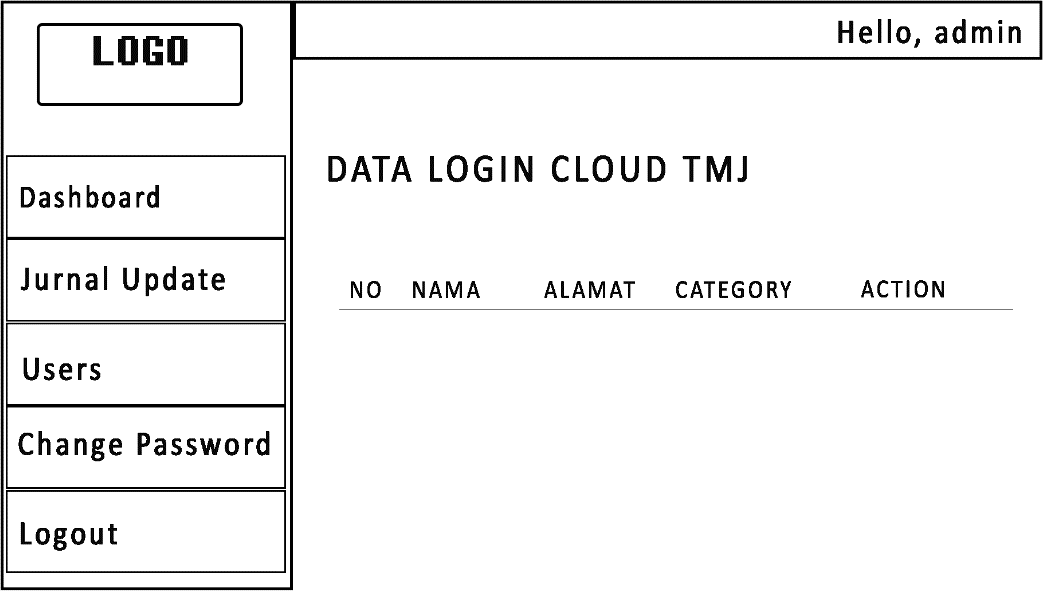
Gambar 3.6 Rancangan Interface Menu Homepage



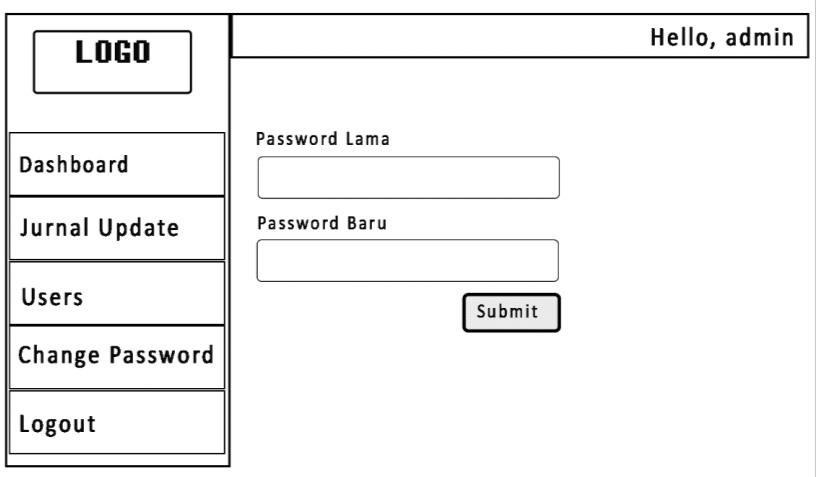
Gambar 3.7 Rancangan Interface Halaman Admin



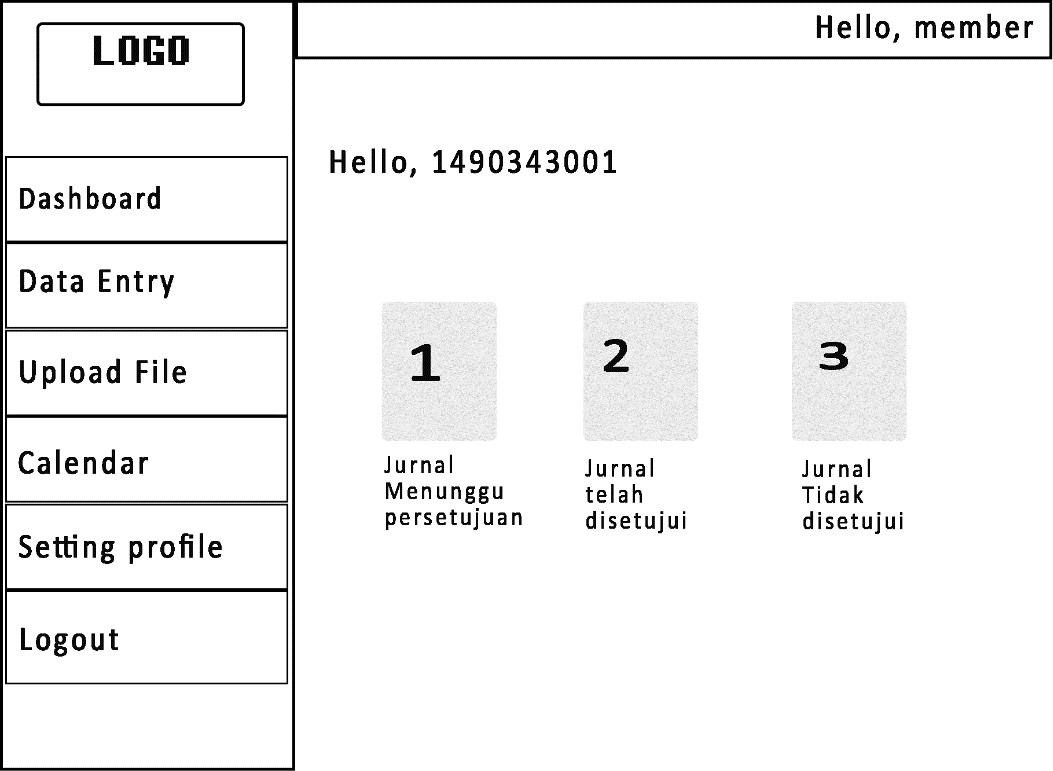
Gambar 3.8 Rancangan Interface Halaman Jurnal Update



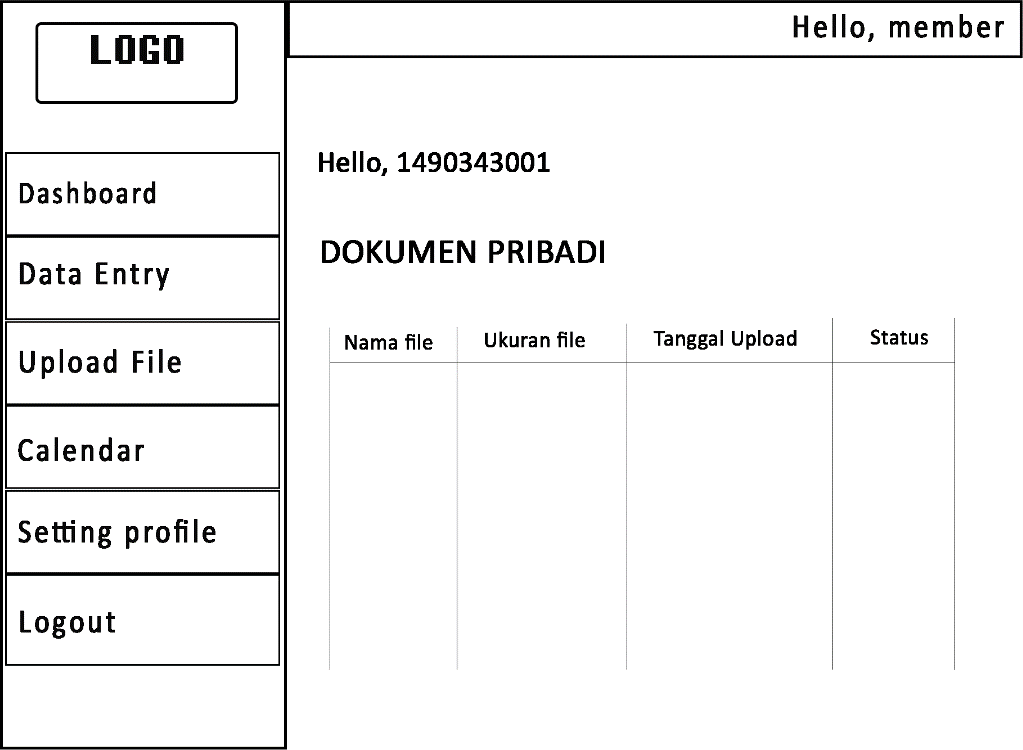
Gambar 3.9 Rancangan Interface Halaman User



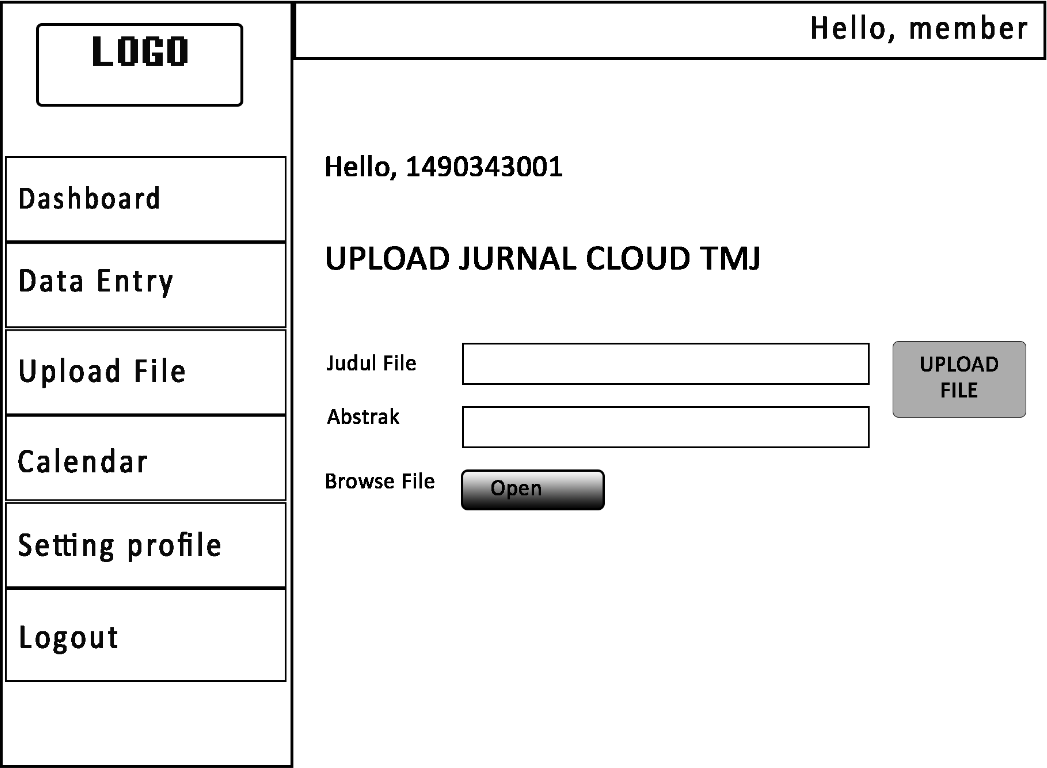
Gambar 3.10 Rancangan Interface Halaman Change Password



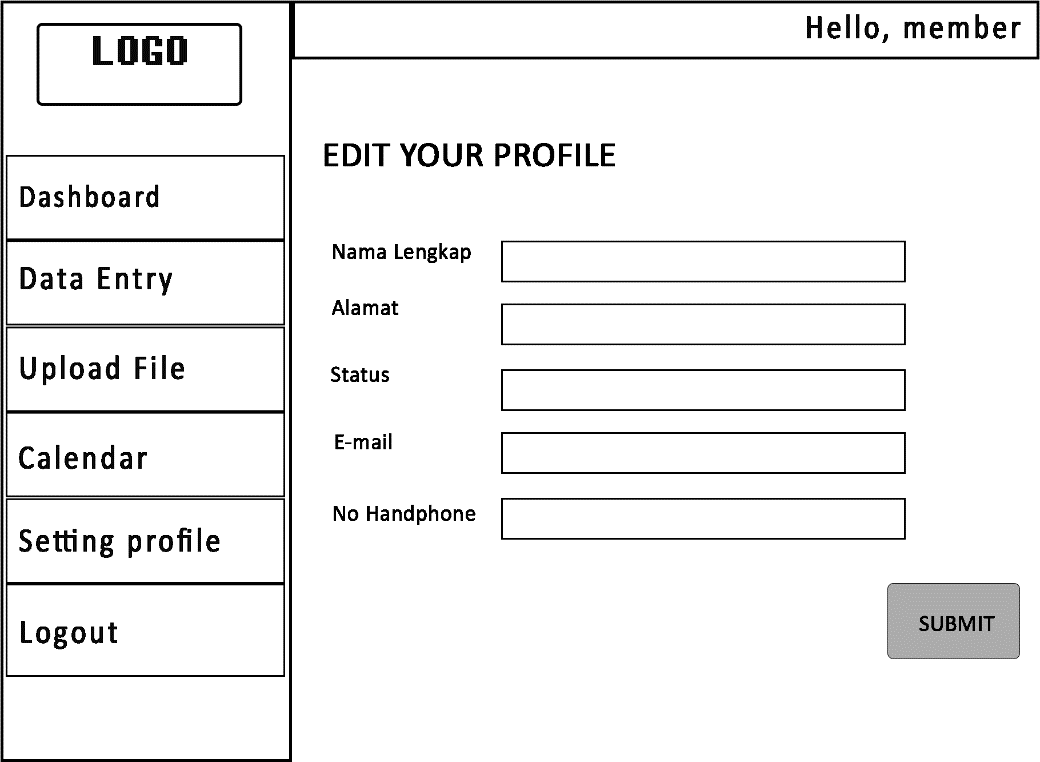
Gambar 3.11 Rancangan Interface dashboard member



Gambar 3.12 Rancangan Interface Halaman Data entry



Gambar 3.13 Rancangan Interface Halaman Upload File



Gambar 3.14 Rancangan Interface Halaman Setting profile

# Skenario Pengujian

* + 1. **Implementasi Algoritma Knuth Morris Pratt**

Sistem yang dibuat akan dilakukan pengujian bertujuan untuk menperkirakan pencocokan string matching mesin pencarian secara manual dari Web layanan *Repository.* Pengukuran performansi merupakan salah satu upaya dalam peningkatan efisiensi dan efektifitas kerja suatu jaringan guna meningkatkan produktifitas kerja pada jaringan tersebut. Untuk menguji pencocokan secara manual dari sistem layanan *Web repository* yang dibangun maka dilakukanlah perhitungan pegeseran setiap karakter tiap-tiap string hingga menemukan kecocokan dai pattern dari text yang telah adalah dalam sistem. Berikut langkah-

langkah yang dilakukan algoritma *Knuth Morris Pratt* pada saat mencocokkan string adalah sebagai berikut:

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

START

Input Kata Kunci yang akan dicari

Keyword ditemukan ?

TIDAK

YA

YA

Kembali melakukan pencarian ?

TIDAK

FINISH

Seleksi Keyword

Menampilkan hasil pencarian

Hitung Jumlah String yang ditemukan pada sistem

Mencari Keyword ke dalam sistem database

Gambar 3.15 *Flowchart* algoritma KMP

# BAB IV

**HASIL DAN PEMBAHASAN**

* 1. **Halaman User Interface**

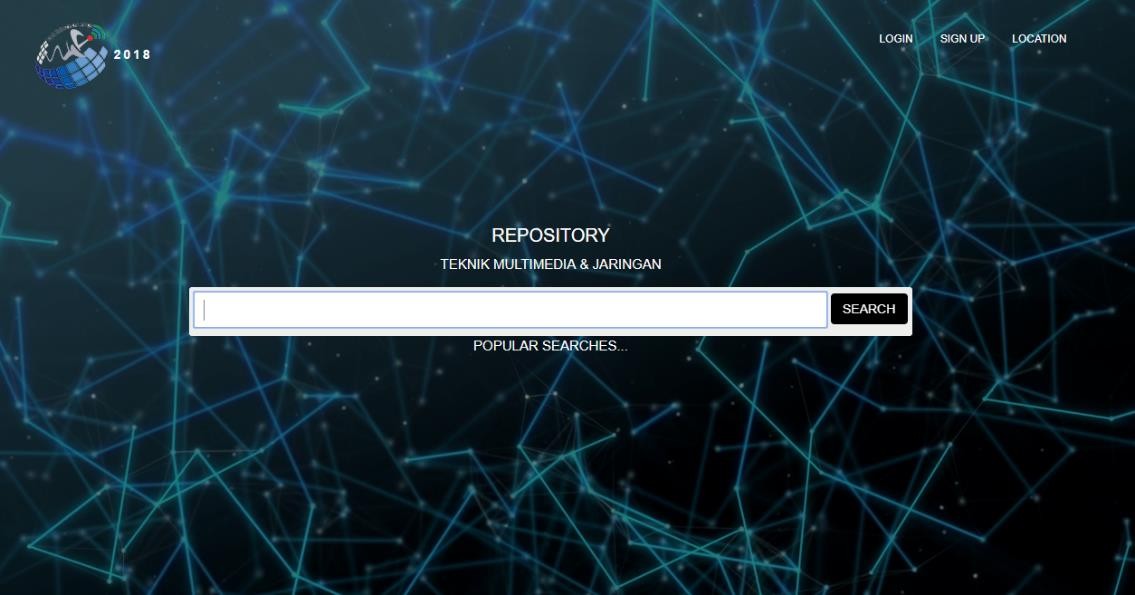
*User interface* merupakan antarmuka yang digunakan oleh *user* atau pengguna agar dapat berinteraksi dengan system baik secara *intranet* maupun *internet*. Pada halaman utama web *user interface* pada sistem ini terdiri dari *home, login, help, about, register, dashboard admin, dashboard member, upload, download*, dan jurnal atau e-book. Pada *dashboard member* dilakukan pengujian pada menu ini member dapat menginput atau menggunggah file dokumen dan terdapat data masuk baik yang belum atau yang sudah disetujui serta dapat mengupdate data biodata sesuai keinginan. Dan pada *dashboard admin* terdapat menu *master user* atau member dimana admin wajib men-verifikasi setiap data registrasi yang masuk dan juga mempunyai hak untuk mempublikasi setiap file dokumen yang akan dipublikasi kelaman utama. Penjelasan lebih lengkap dan diuraikan sebagai berikut

# Tampilan Homepage

*Homepage* merupakan halaman awal atau beranda yang muncul saat pertama kali member dan admin menjalankan web, dan halaman ini bias diakses oleh siapa saja. Pada halaman utama memiliki beberapa menu dan fungsi yang terdapat pada sistem layanan *repository* ini yaitu *home, login, help, about, register,*

41

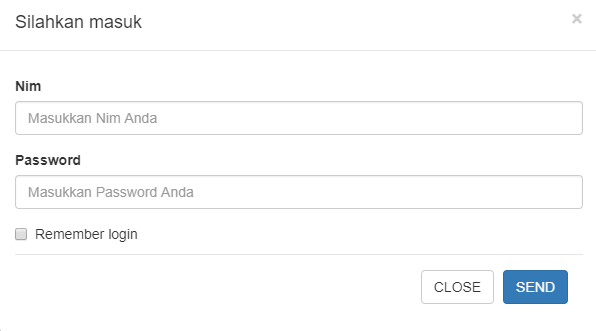
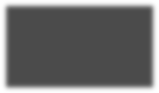
*upload, download*, dan *jurnal* atau *e-book.* Tampilan *Homepage* dapat dilihat pada gambar 4.1



Gambar 4.1 Tampilan *Homepage*

# Tampilan Form Login

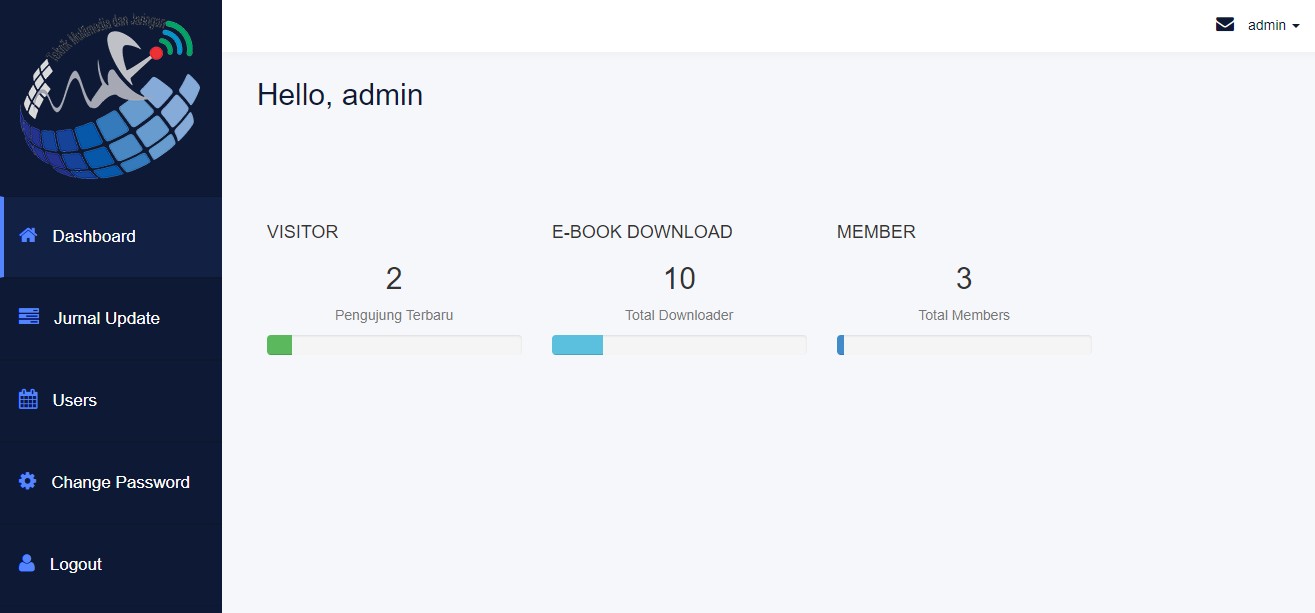
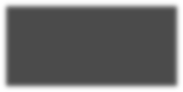
Menu login berfungsi sebagai form untuk admin dan member dapat mengakses halamannya tesendiri, admin akan dimintai agar mengisi username dan password supaya dapat masuk ke halaman yang dituju sesuai akses user. Menu login member juga diminta untuk menginput data username dan password agar bias mengakses laman tersendiri dengan maksud untuk mencegah pihak yang tidak berwewenang masuk mengakses *system* dan berusaha untuk merusaknya. Tampilan form login data dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan Form

# Dashboard Utama Admin

Pada halaman utama admin merupakan tampilan awal saat melakukan *login* dan halaman ini hanya bias diakses oleh admin, dan terdapat beberapa menu yaitu menu *dashboard admin* yang pertama kali tampil setelah admin melakukan *login,* menu jurnal update yaitu menu data masuk jurnal dari proses upload member, menu *users* merupakan menu data menajemen *user*, dan terakhir menu *change password* untuk mengubah password admin saat *login*. Dan berikut merupakan tampilan dashboard admin, dapat dilihat pada gambar 4.3.



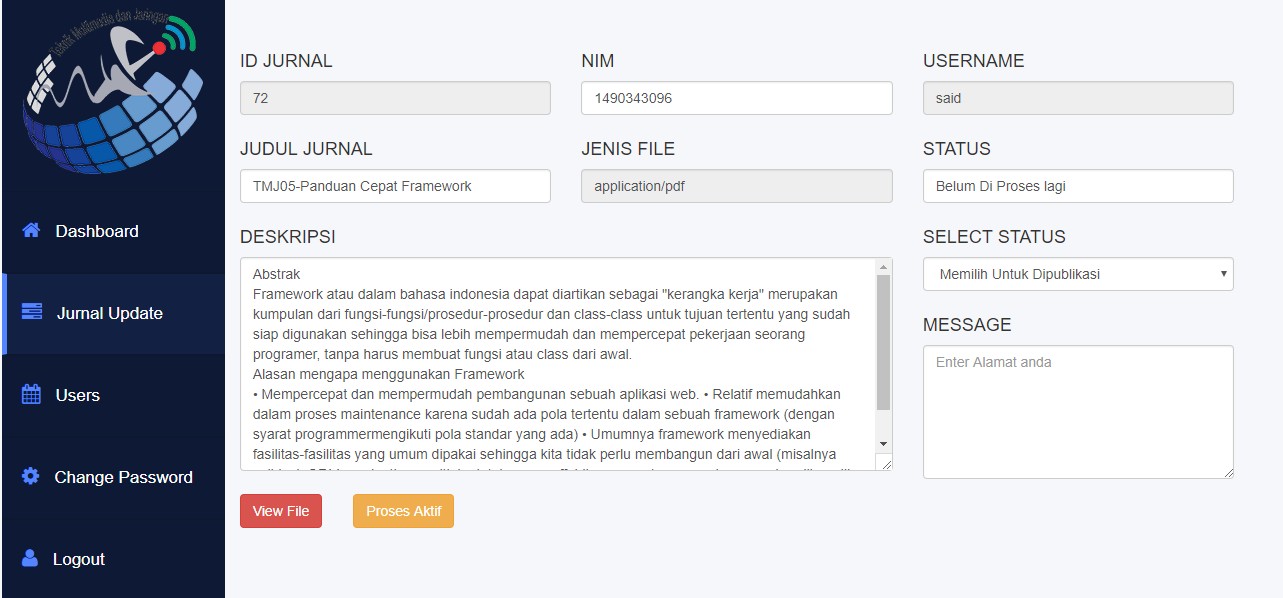
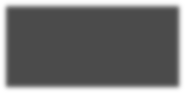
Gambar 4.3 Dashboard Utama Admin

Pada dashboard admin terdapat menu jurnal update yaitu menu data informasi jurnal dari semua member, pada menu ini admin dapat melihat deskripsi jurnal dan info mengenai jurnalnya, admin diberikan hak untuk menyetuju setiap jurnal yang masuk untuk dapat dipublikasi pada halaman menu utama website. Berikut tampilan menu jurnal update, dapat dilihat pada gambar 4.4.



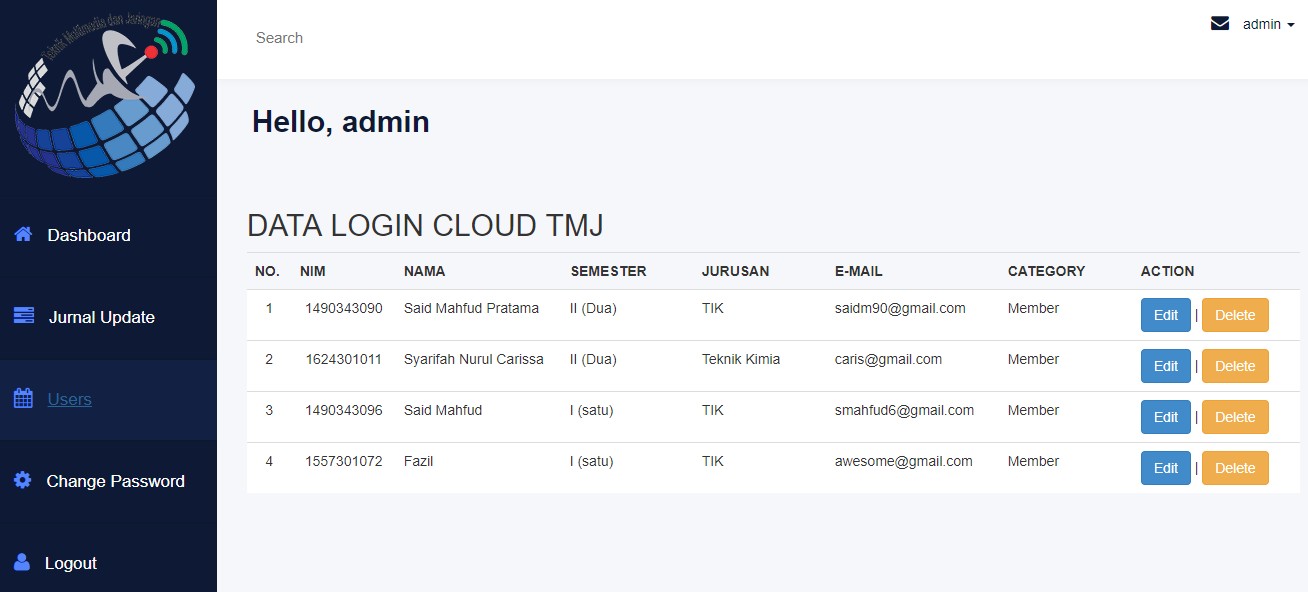
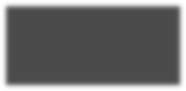
Gambar 4.4 Tampilan Jurnal Update

Pada halaman menu jurnal update admin dapat memilih proses aktif untuk setiap jurnal yang akan diposting di laman *homepage* agar dapat diunduh oleh semua orang yang mengunjungi website, dan admin juga dapat menginput pesan. Berikut tampilan *view detail*, dapat dilihat pada gambar 4.5.



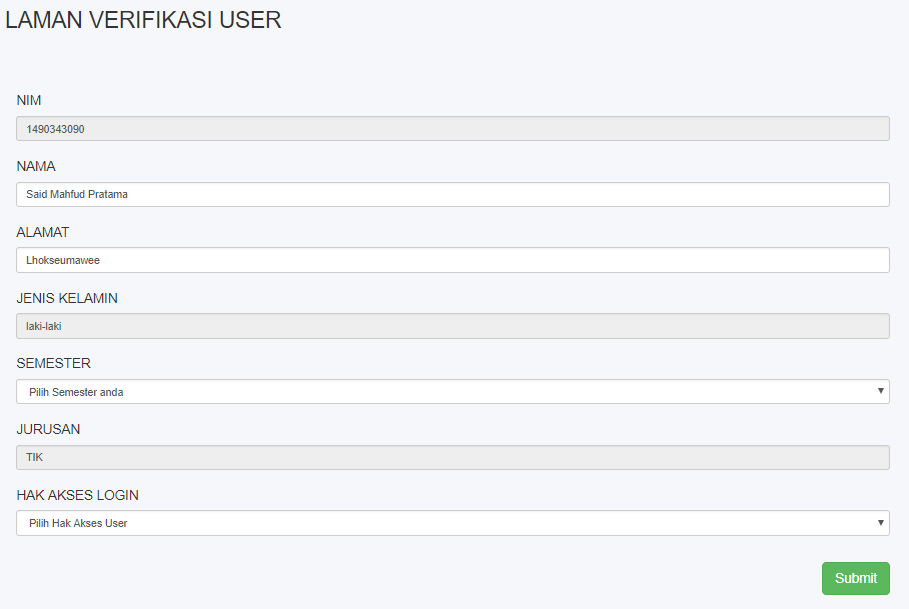
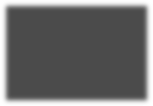
Gambar 4.5 Tampilan *View Detail*

Pada halaman *dashboard admin* terdapat menu *users* yaitu menu data masuk biodata dari setiap member yang registrasi, pada menu admin dapat mengedit kategori *user* yang terbagi menjadi admin ataupun member, dan admin juga bisa menghapus setiap data yang kurang jelas. Berikut tampilan menu *user*, dapat dilihat pada gambar 4.6.



Gambar 4.6 Tampilan Menu User

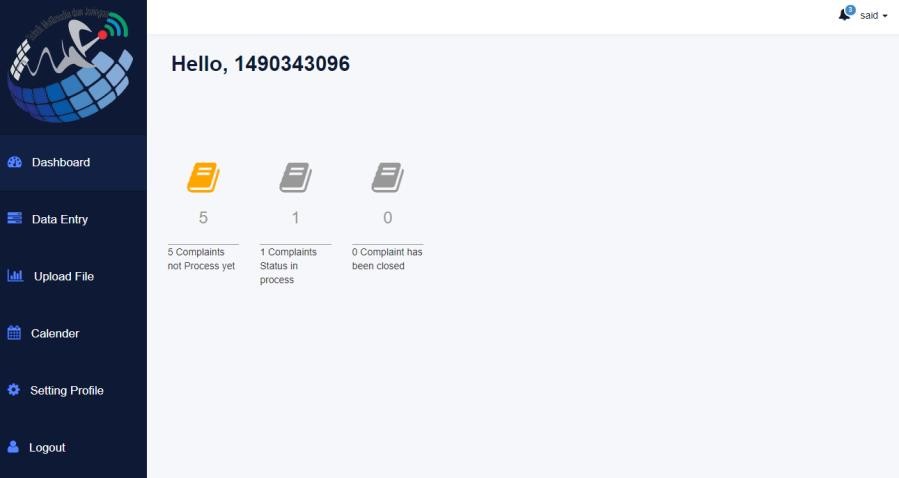
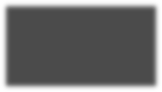
Pada halaman menu *users*, terdapat laman untuk edit hak akses *user* yang berhak dilakukan oleh admin saja. Berikut tampilan data *user*, dapat dilihat pada gambar 4.7.



Gambar 4.7 Tampilan Menu Edit *User*

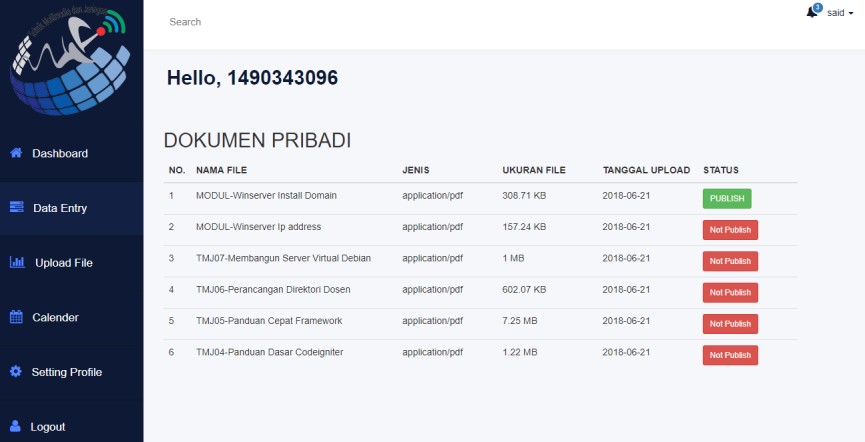
# Dashboard Utama Member

Dashboard utama member adalah tampilan awal member ketika *login* masuk setelah diverifikasi admin, dan halaman ini hanya dapat diakses oleh member saja. Pada halaman ini member dapat mengunggah file dokumen, dan *data entry*, pada halaman ini juga member dapat mengupdate data pribadi, berikut tampilan dashboard utama member, dapat dilihat pada gambar 4.8.



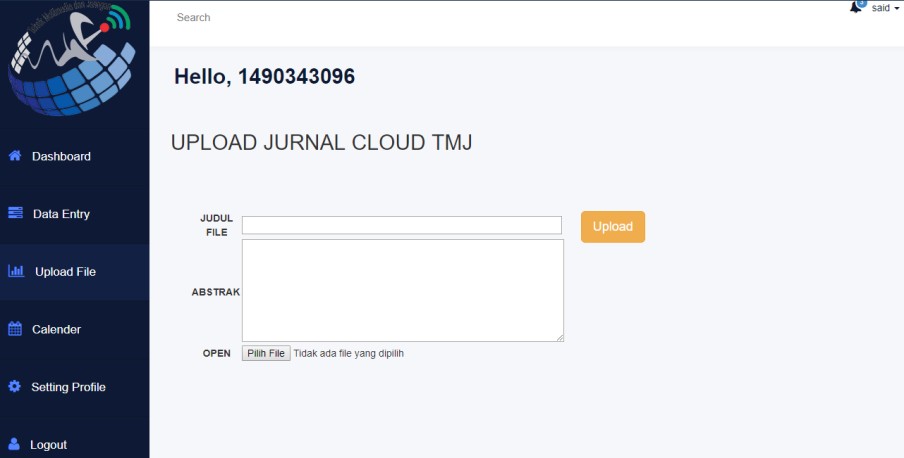
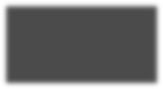
Gambar 4.8 Tampilan Dashboard Member

Pada dashboard member, member dapat melihat data dokumen yang telah di upload baik yang sudah di publikasi maupun yang belum, data masuk akan menampilkan info detail. Berikut tampilan menu *data entry*, dapat dilihat pada gambar 4.9.



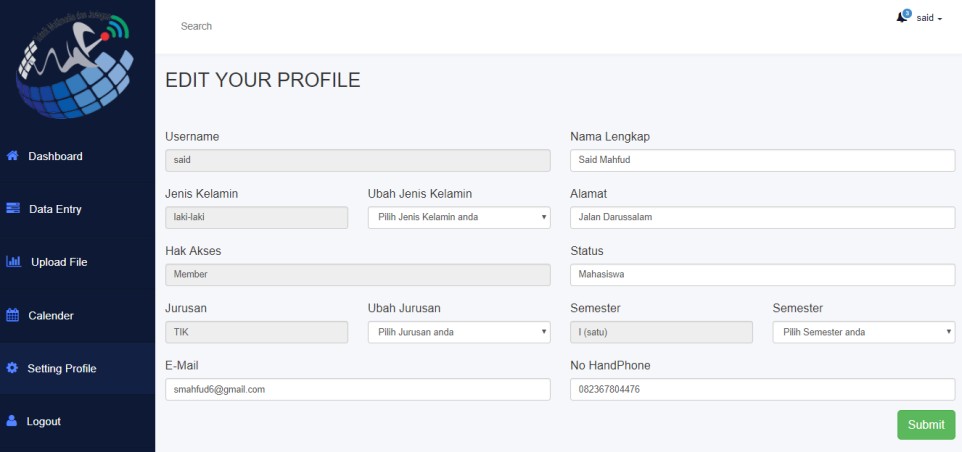
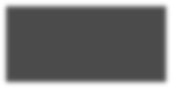
Gambar 4.9 Tampilan *Data Entry*

Halaman selanjutnya adalah proses mengunggah file dokumen member, dimana terdapat kolom judul file, abstrak atau deskripsi, dan tombol *open file*. Member dapat melakukan proses menggunggah data file dan dokumen dalam format .doc dan .pdf yang maksimal ukurannya 10 mb pada sistem layanan *repository*. Berikut merupakan tampilan halaman input file dokumen, dapat dilihat pada gambar 4.10.



Gambar 4.10 Tampilan *Upload File*

Menu selanjutnya merupakan biodata dari member yang berisikan informasi lengkap dan meng-update data pribadi yang terlihat dinamis. Menu ini juga sudah dibuat agar *primary key* tidak dapat diubah. Berikut tampilan *setting profile*, dapat dilihat pada gambar 4.11.



Gambar 4.11 Tampilan *Setting Profile*

# Tampilan Help

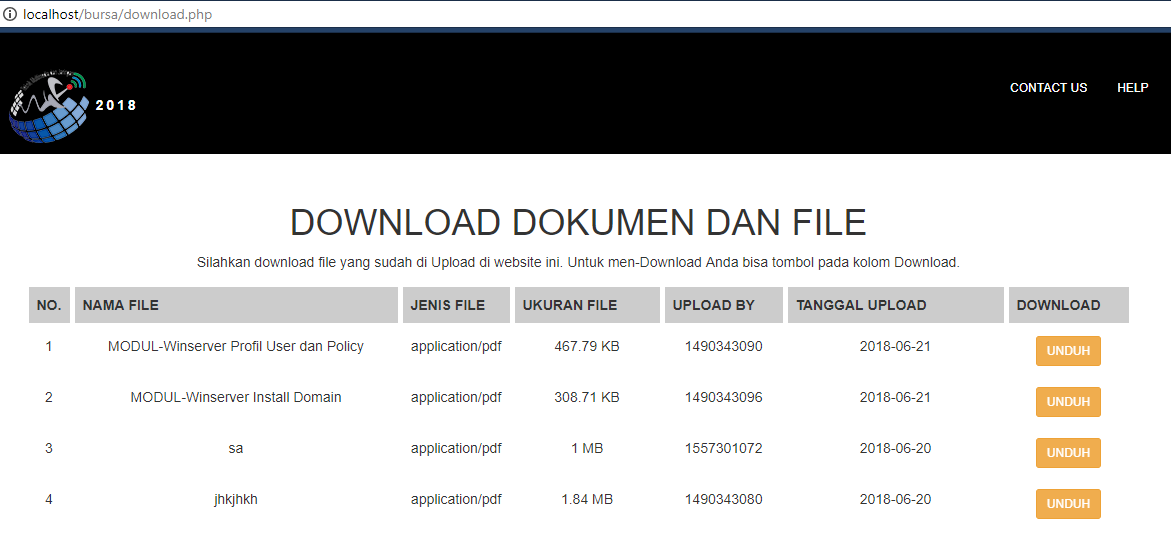
*Form help* merupakan menu yang berisikan petunjuk dan intruksi kepada *user* bagaimana cara menjalankan dan memanfaatkan sistem layanan *repository* program studi Teknologi Rekayasa Komputer Jaringan. Selengkapnya tampilan *help* dapat dilihat pada gambar 4.12 berikut ini.

Gambar 4.12 Tampilan *Help*



# Tampilan Menu Download

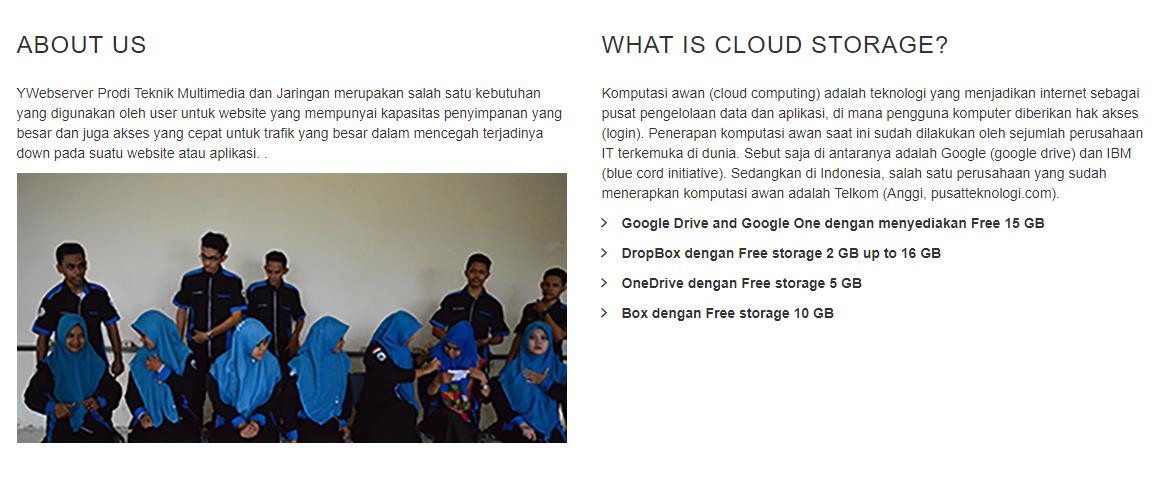
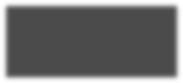
Form Download merupakan form untuk menampilkan hasil file yang telah di setujui oleh admin. Dari beberapa member yang sudah megunggah file jurnal yang dapat terlihat pada *home website*. Berikut merupakan tampilan form download pada gambar 4.13.



Gambar 4.13 Tampilan *Form Download*

# Tampilan Menu About

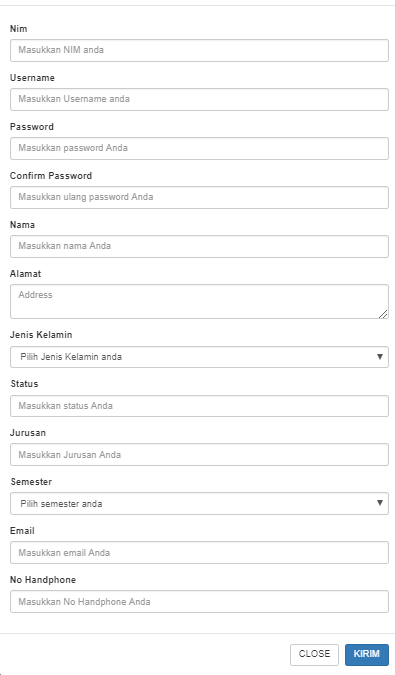
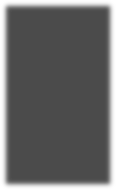
Pada menu about berisi tentang informasi dasar dari *Website repository* prodi Teknologi Rekayasa Komputer Jaringan. Menu about dapat dilihat pada gambar 4.14 berikut ini.



Gambar 4.14 Tampilan *Form About*

# Tampilan Registrasi

Pada tampilan register merupakan proses untuk menyimpan data *user* pada *database* agar mendapatkan hak akses *user* pada sistem yang telah dibangun di layanan *repository* ini. Berikut bentuk form registrasi *user* baru pada gambar 4.15.

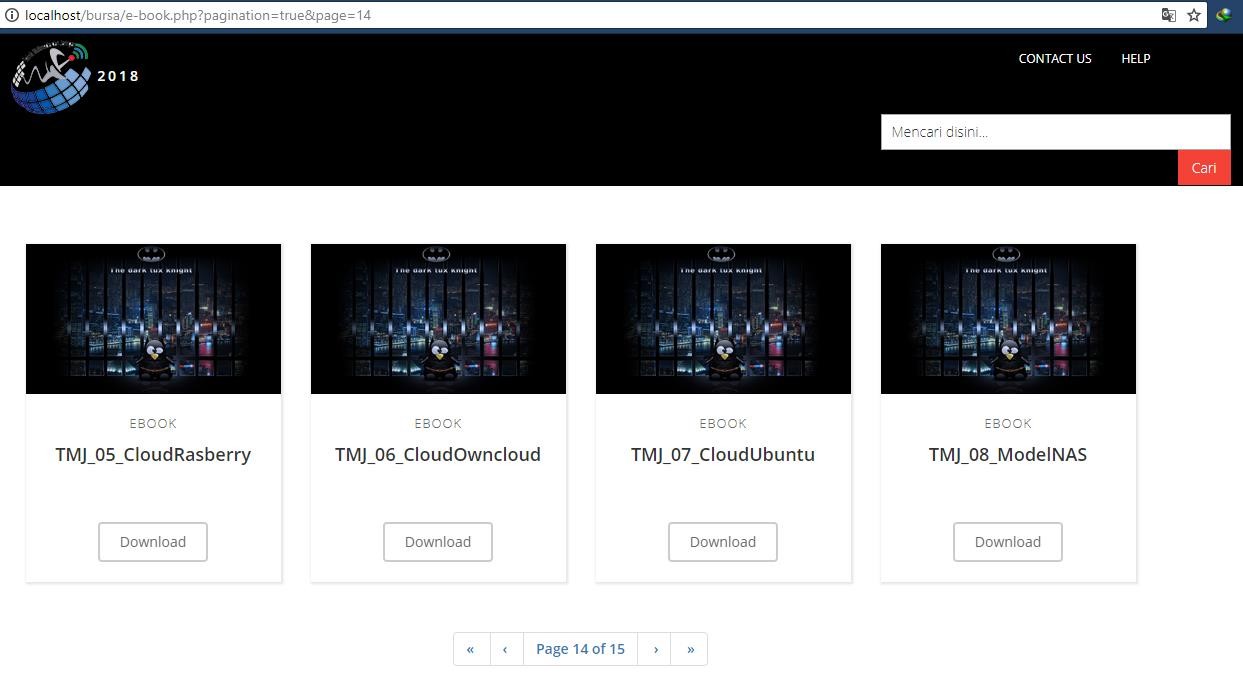


Gambar 4.15 Tampilan Register

# Tampilan E-book

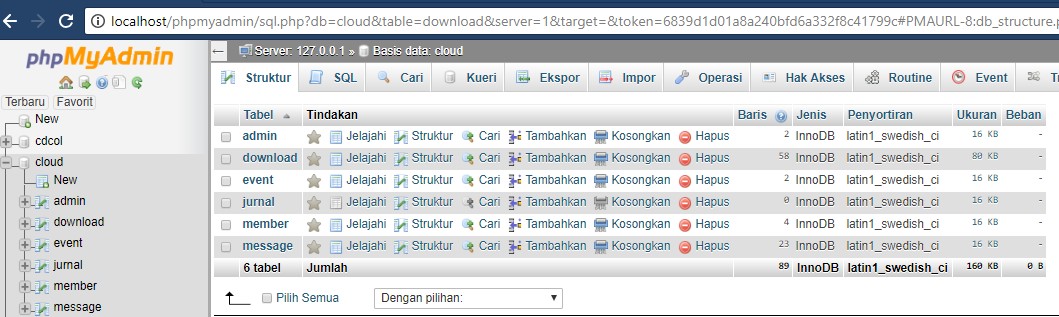
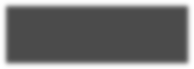
Pada tampilan *e-book* merupakan kumpulan beberapa *e-book* yang terhubung dengan *database* dan bisa langsung di unduh. Dan untuk tampilan jurnal akan menampilkan jurnal ter-update dan dapat melihat *preview file* tersebut. Berikut dapat dilihat pada gambar 4.16.

Gambar 4.16 Tampilan *e-book*



# Tampilan Database

Pada website ini *database* dibuat menggunakan *database* phpmyadmin yang berfungsi untuk menyimpan data. Untuk melakukan penginputan atau melihat data yang ada dalam *database* dapat mengakses dengan menggunakan aplikasi *web server* Xampp dan mengaktifkan fitur apache dan mysql kemudian pada *address bar browser*, akses localhost/phpmyadmin/ dan hasilnya database yang telah pernah dibuat akan muncul. Berikut gambar 4.17 merupakan tampilan *database cloud.sql* yang diakses melalui salah satu browser.



Gambar 4.17 Tampilan *Database*

* 1. **Implementasi Metode *Searching Knuth Morris Pratt* (KMP)**

Pada bagian penerapan metode *searching*, algoritma *Knuth-Morris- Pratt* merupakan salah satu algoritma pencarian string (*string matcing*) dimana ketika pencarian dilakukan maka algoritma akan mulai mencocokkan karakter *pattern* atau pola pada awal teks. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks (*keyword*) yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:

1. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (mismatch).
2. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan *alert* penemuan di posisi ini dan berapa jumlah kata yang ditemukan.
3. Algoritma kemudian menggeser *pattern* berdasarkan *table next* (lompat), lalu mengulangi langkah 2 sampai *pattern* berada di ujung atau akhir teks.

Berikut adalah *listing program* algoritma *Knuth Morris Pratt* sebagai metode pencaharian.

/\* menetukan tabel lompatan dengan preKMP

$i = $lompat[$i];

}

$i++;

$j++; if($i>=count($pattern)){

// jika cocok, tentukan posisi string yang cocok

// kemudian lompat ke string berikutnya

$hasil[$num++]=$j-count($pattern);

$i = $lompat[$i];

}

}

return $hasil;

}

cocok, maka lompat sesuai tabel

tidak

// jika

lompatan

<?php error\_reporting(0); class KMP{

function KMPSearch($p,$t){

$hasil = array();

$pattern = str\_split($p);

$text = str\_split($t);

$lompat = $this->preKMP($pattern);

//print\_r($lompat);

// perhitungan KMP

$i = $j = 0;

$num=0; while($j<count($text)){

while($i>-1 && $pattern[$i]!=$text[$j]){

dengan str2

* $str2 = (array string) string yang akan mengganti

str1

* $text = (string) text yang akan dicari
* output :
* $t = teks yang sudah difilter

\*/

function KMPReplace($str1,$str2,$text){

$str1 = (array string) string yang akan diganti

\*

/\* replace string

\* input :

* input :
* $pattern = (string) pattern
* output :
* $lompat = (array int) untuk jumlah lompatan

\*/

function preKMP($pattern){

$i = 0;

$j = $lompat[0] = -1; while($i<count($pattern)){

while($j>-1 && $pattern[$i]!=$pattern[$j]){

$j = $lompat[$j];

}

$i++;

$j++; if($pattern[$i]==$pattern[$j]){

$lompat[$i]=$lompat[$j];

}else{

$lompat[$i]=$j;

}

}

return $lompat;

}

$num = 0;

$location = $this->KMPSearch($str1,$text);

$t = '';

$n = 0; $nn = 0; foreach($location as $in){

$t .= substr($text,$n+$nn,$in-$n-$nn).$str2;

$nn = strlen($str1);

$n = $in;

}

$t .= substr($text,$n+$nn); return $t;

}

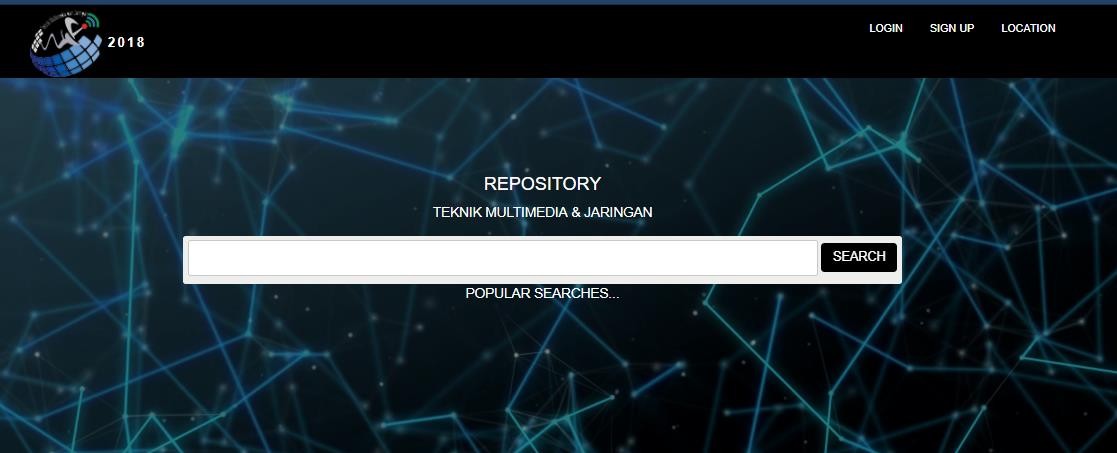
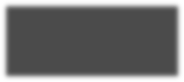
}

?>

# Pengujian Algoritma Knuth Morris Pratt (KMP)

Pengujian sistem merupakan proses yang dilakukan untuk mengetahui seberapa jauh Website ini telah berjalan. Pada tahap inilah semua proses akan dikerjakan mulai dari proses pencarian tiap *string*, mencocokan antara kata unik dari kumpulan kata yang diinput atau ada dalam *database*.

Pada halaman utama terdapat *search bar* yang dapat dilihat oleh setiap pengunjung. Pada pencarian ini telah menggunakan metode searching *knuth morris pratt.*Berikut merupakan tampilan *search bar* pada *homepage*, dapat dilihat pada gambar 4.18.



Gambar 4.18 Tampilan *Search bar*

Berikut potongan *listing program* untuk koneksi menggunakan pencarian dengan algoritma *Knuth Morris Pratt.*

<?php

$c = mysql\_connect("localhost", "root", "");

$db = mysql\_selectdb("cloud",$c);

include\_once("KMP.php");

$kata = ''; if(isset($\_GET['kata']))

$kata = $\_GET['kata'];

echo "<form class='form-wrapper cf' action='' method='post'>

<input type='text' name='kata' value=$kata

>

<button type='submit' name='cari'><b>Cari</b></button></form>";

$KMP = new KMP();

if(isset($\_POST['cari'])){

$kata = $\_POST['kata'];

$sql ="SELECT \* FROM download WHERE nama\_file LIKE '%$kata%' AND status='PUBLISH'";

$result = mysql\_query($sql);

$jumlah = mysql\_num\_rows($result);

if (empty($kata)){

echo "<script language='javascript'>alert('Anda Belum Memasukkan Judul Tugas Akhir Yang Ingin Dicari!!!'); document.location='koneksi\_kmp.php'</script>";

}

else if ($jumlah>0){

echo "<h2>REPOSITORY TEKNOLOGI REKAYASA KOMPUTER JARINGAN</h2>";

echo "Jumlah judul yang ditemukan : ".$jumlah."<br>";

echo "<b><font color='#cec708'>Pencarian query &nbsp;<i>$kata</i> &nbsp;&nbsp; ditemukan pada judul :

</font></b><br><br>";

while ($teks = mysql\_fetch\_array($result)){

$hasil = $KMP->KMPSearch($kata,$teks['nama\_file']); echo "<div style='width:600px;'>";

echo nl2br(str\_replace($kata,"<font color='red'>".$kata."</font>",$teks['nama\_file']));

echo " <font='blue'><a href='repositori\_lengkap.php?NPM=$teks[abstrak]'>Baca Selengkapnya</a></font>";

echo "<br><br/>"; echo "</div>";

}

} else{

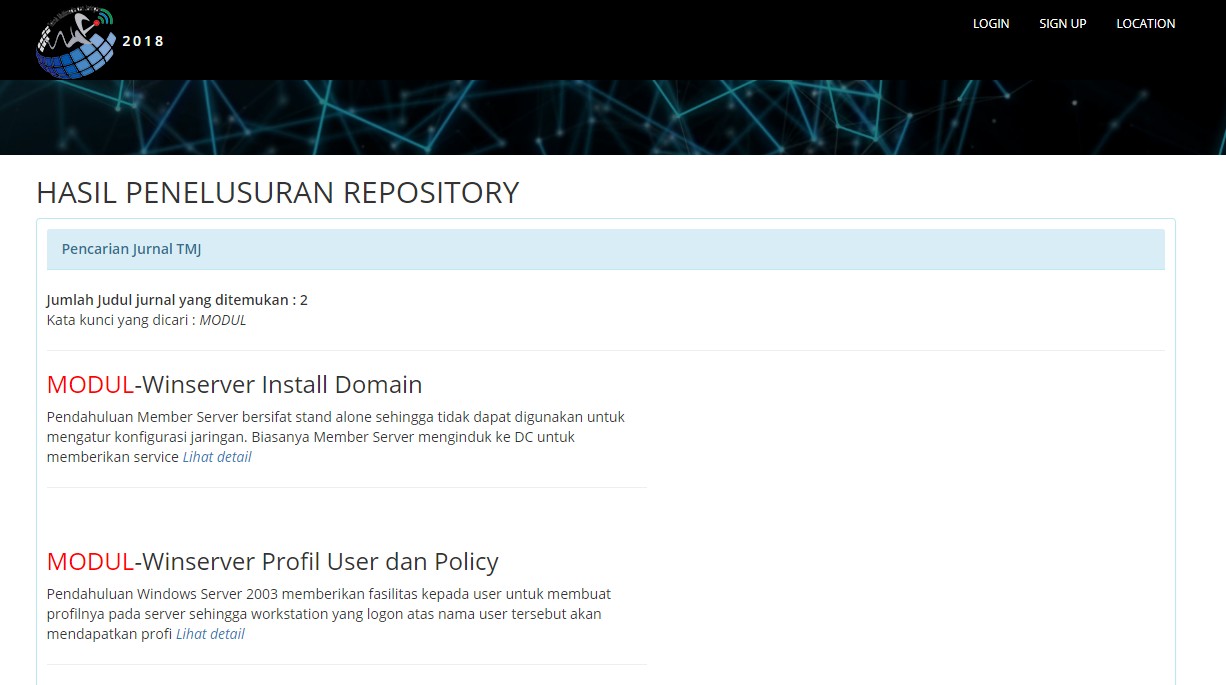
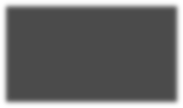
echo "JUDUL YANG ANDA MAKSUDCARIKAN TIDAK ADA!!";

}

}

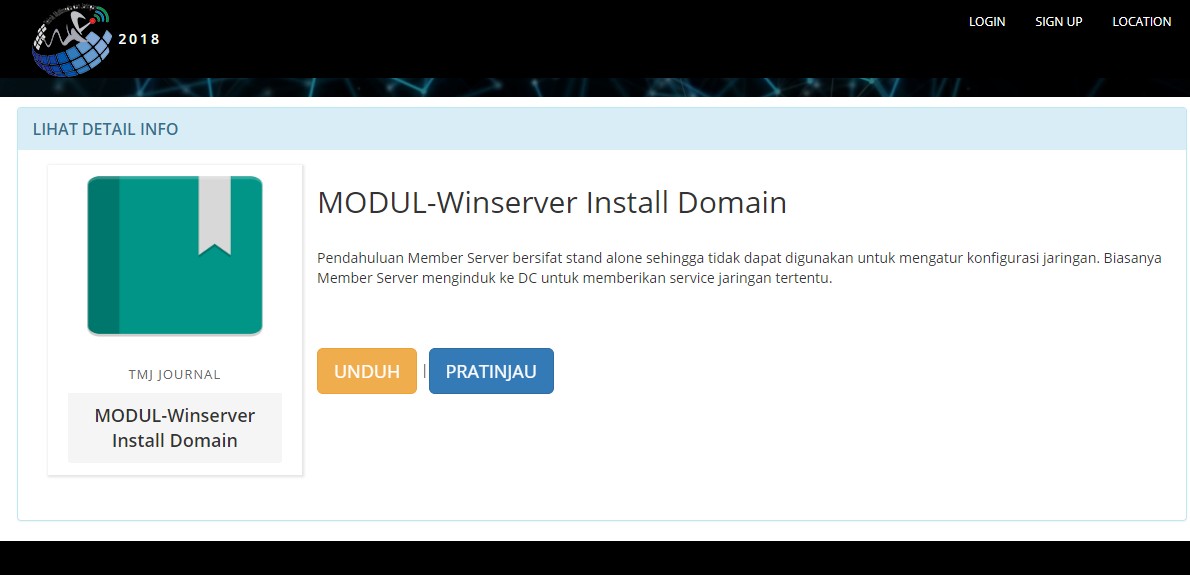
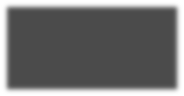
?>

Pada Menu *Homepage* yang berada pada halaman utama website, data akan muncul ketika pencaharian mulai melakukan proses dan semua orang dapat mengaksesnya tanpa harus melakukan tahapan registrasi terlebih dahulu. Pada menu *homepage* ini data akan tampil sesuai dengan inputan kata yang dimasukkan atau dicari. Berikut merupakan proses hasil penelusuran kata, dapat dilihat pada gambar 4.19.



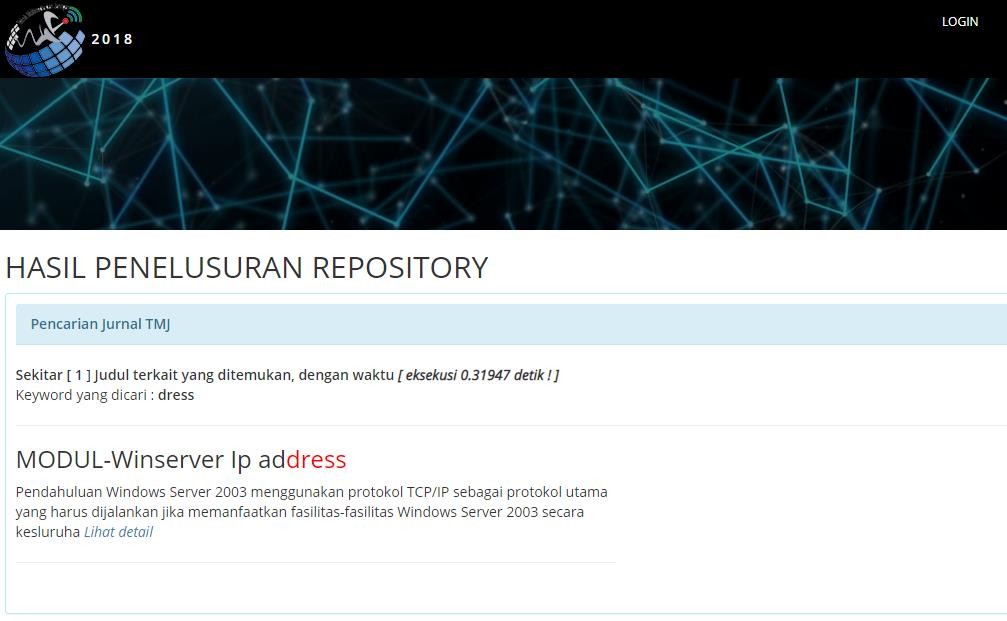
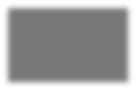
Gambar 4.19 Tampilan Hasil Penelusuran awal

Kemudian jika ingin lihat *info detail* akan tampil seperti gambar 4.20 dibawah ini



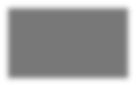
Gambar 4.20 Tampilan Detail Penelusuran

Berikut merupakan percobaan pencarian kata berdasarkan beberapa kriteria yang telah diuji pada system dan terdapat data waktu eksekusi yang dihasilkan, diantara lain terdapat pada gambar 4.21, gambar 4.22 dan gambar 4.23 dibawah.



Gambar 4.21 Hasil Penelusuran karakter kata

Gambar 4.22 Hasil Penelusuran 2 kata dalam satu judul teks



Gambar 4.23 Hasil Penelusuran 1 kata dalam dua judul teks berbeda

# Pengujian Sistem Secara Manual

* + 1. Pembuktian Manual Pencarian Kata

Pada saat melakukan proses pencarian jurnal atau file dokumen, pengunjung atau visitor website hendak ingin mencari jurnal atau dokumen yang hendak akan dibaca. Misalkan ada kasus pengunjung mencari judul, kata kunci yang akan dicari adalah “ADDRESS”, dan pattern yang akan dicari adalah “DRESS”. Pengguna ingin mencari jurnal secara tepat, dan memilih algoritma untuk pencarian text tersebut.

Penyelesaian untuk kasus diatas jika menggunakan algoritma *Knuth Morris Pratt* adalah sebagai berikut:

1. Terlebih dahulu menentukan *pattern* dan teks yang akan dicari. Dalam kasus ini, pattern dan teks yang hendak dicari adalah sebagai berikut:

P = DRESS

T = ADDRESS

1. Selanjutnya menentukan fungsi pinggiran dari pattern dan teks yang akan dicari.

Fungsi pinggiran disini didefinisikan sebagai ukuran awalan terpanjang dari pattern. Fungsi pinggiran tersebut telah dijabarkan dalam Tabel 4.1, seperti dibawah ini.

Tabel 4.1 Fungsi Pinggiran kasus pencarian *string*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J | 0 | 1 | 2 | 3 | 4 | 5 | 6 | n.. |
| p (j) | A | D | D | R | E | S | S |  |
| b (j) | 0 | 1 | 3 | 0 | 0 | 0 | 0 |  |

1. Memberikan nilai indeks atau lakukan peng-index-an ke *pattern* dan teks.

Adapun nilai indeks pattern dan teks akan dipaparkan dalam table 4.2 dan 4.3 seperti berikut.

Tabel 4.2 Nilai indeks untuk Teks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| INDEKS | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| t (j) | A | D | D | R | E | S | S |

Tabel 4.3 Nilai indeks untuk *Pattern*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INDEKS | 0 | 1 | 2 | 3 | 4 |
| p (j) | D | R | E | S | S |

1. Cara perhitungan pergeseran dari algoritma *Knuth-Morris-Pratt* adalah sebagai berikut :
   1. Membandingkan ujung kiri pada teks dan pada pattern.

Pada ujung kiri teks dan ujung kiri *pattern* terjadi ketidakcocokan, tetapi pada nilai indeks ke 3, terjadi kecocokan. Teks dengan karakter indeks pertama A sedangkan pattern ketiganya yaitu D.

* 1. Karena terjadi ketidakcocokan, maka lakukan pergeseran *pattern* P dengan jumlah pergeseran sesuai dengan nilai pinggiran pattern p yang cocok. Pada kasus ini, karakter pada *pattern* dan teks yang menemukan kesamaan ada di indeks 2-6, maka panjang kesesuaian teks adalah 5 (*l=5*).
  2. Nilai pinggiran terpanjang dari *pattern* P yang menemukan kecocokan adalah P[2…6], dimana dalam fungsi pinggiran sebelumnya, nilai *pattern* pada indeks urutan 3 dalam fungsi pinggiran adalah 3 (b(3) =()).
  3. Setelah ditentukan nilai fungsi pinggiran dan panjang dari kecocokan teks dan *pattern*, maka pergeseran karakter dilakukan dengan cara :

Nilai Pergeseran = *l – b*

Keterangan :

Nilai pergeseran : Nilai yang digeser tiap karakternya

I = Panjang Kecocokan karakter antara *pattern* dan teks b = Nilai dalam fungsi pinggiran

Maka, Nilai Pergeseran = *l – b*

= 5 – 3 = 2 karakter

Jadi, pattern P digeser sebanyak 2 karakter ke kanan. Pengeseran tersebut akan terlihat pada table 4.4, sebagai berikut.

Tabel 4.4 Nilai indeks untuk *Pattern*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEKS TEKS T |  | 0 | 1 |  | 2 | 3 | 4 | 5 | 6 |
| TEKS |  | A | D |  | D | R | E | S | S |
| INDEKS PATTERN P | PERGESERAN | | | | 0 | 1 | 2 | 3 | 4 |
| PATTERN |  | | | | D | R | E | S | S |

* + 1. Pembuktian Pencarian Kata yang sama dalam satu Judul File

Pada saat melakukan proses pencarian jurnal atau file dokumen, misalkan ada kasus pengunjung mencari judul, teks tujuan yang akan dicari adalah “XUBUNTU DAN LUBUNTU”, dan pattern yang akan dicari adalah “BUNTU”. Pengguna ingin mencari jurnal secara tepat, dan memilih algoritma untuk pencarian text tersebut.

Penyelesaian untuk kasus diatas jika menggunakan algoritma Knuth morris pratt adalah sebagai berikut:

1. Terlebih dahulu menentukan pattern dan teks yang akan dicari. Dalam kasus ini, pattern dan teks yang hendak dicari adalah sebagai berikut:

P = BUNTU

T = XUBUNTU DAN LUBUNTU

1. Selanjutnya menentukan fungsi pinggiran dari pattern dan teks yang akan dicari.

Fungsi pinggiran disini didefinisikan sebagai ukuran awalan terpanjang dari pattern. Fungsi pinggiran tersebut telah dijabarkan dalam Tabel 4.5, seperti dibawah ini.

Tabel 4.5 Fungsi Pinggiran kasus pecarian *string*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | n.. |
| p(j) | X | U | B | U | N | T | U |  | D | A | N |  | L | U | B | U | N | T | U |  |
| B(j) |  |  |  |  |  |  |  |  |  | 0 | 0 |  |  | 0 | 0 | 0 | 0 | 0 | 0 |  |

1. Memberikan nilai indeks atau lakukan peng-index-an ke pattern dan teks.

Adapun nilai indeks pattern dan teks akan dipaparkan dalam table 4.6 dan 4.7 seperti berikut.

Tabel 4.6 Nilai indeks untuk Teks

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEKS | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| t (j) | X | U | B | U | N | T | U |  | D | A | N |  | L | U | B | U | N | T | U |

Tabel 4.7 Nilai indeks untuk Pattern

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INDEKS | 0 | 1 | 2 | 3 | 4 |
| p (j) | B | U | N | T | U |

1. Cara perhitungan pergeseran dari algoritma *Knuth-Morris-Pratt* adalah sebagai berikut :
   1. Membandingkan ujung kiri pada teks dan pada pattern.

Pada ujung kiri teks dan ujung kiri pattern terjadi ketidakcocokan, tetapi pada nilai indeks ke-3 dan ke-15, terjadi kecocokan. Teks dengan karakter indeks pertama X sedangkan pattern ke-3 dan pattern ke-15 yaitu B.

* 1. Karena terjadi ketidakcocokan, maka lakukan pergeseran pattern P dengan jumlah pergeseran sesuai dengan nilai pinggiran pattern p yang cocok. Pada kasus ini, karakter pada pattern dan teks yang menemukan kesamaan ada di indeks 2-6 dan juga pada indeks 14-18, maka panjang rata-rata kesesuaian teks adalah 5 (*l=5*).
  2. Nilai pinggiran terpanjang dari pattern P yang menemukan kecocokan adalah P1[2…6] dan P2[14…18], dimana dalam fungsi pinggiran sebelumnya, nilai pattern1 pada indeks urutan 3 dalam fungsi pinggiran adalah 3 (b (3) =(3)) sedangkan nilai pattern2 pada indeks urutan ke-15 dalam fungsi pinggiran adalah 15(b (15) =(15)).
  3. Setelah ditentukan nilai fungsi pinggiran dan panjang dari kecocokan teks dan pattern, maka pergeseran karakter dilakukan dengan cara :

Nilai Pergeseran = *l – b*

Keterangan :

Nilai pergeseran : Nilai yang digeser tiap karakternya

*I* = Panjang Kecocokan karakter antara pattern dan teks b = Nilai dalam fungsi pinggiran

Maka, Nilai Pergeseran (1) = *I – b*

= 5 – 3 = 2 karakter

Jadi, pattern P digeser sebanyak 2 karakter ke kanan untuk pencarian kata. Pengeseran tersebut akan terlihat pada table 4.8, sebagai berikut.

Tabel 4.8 Hasil pergeseran 2 karakter kata teks dan pattern

Pattern

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEKS Teks | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 18 |
| TEKS | X | U | B | U | N | T | U |  | D | A | N |  | L | U | B | U | N | T U |
| INDEKS |  | PER | 2 | 3 | 4 | 5 | 6 |  |  |  |  |  |  |  | 14 | 15 | 16 | 17 18 |

PERGESERAN

TEKS B U N T U B U N T U

GESERAN

* + 1. Pembuktian Pencarian Kata yang sama dalam 2 Judul berbeda

Berikut dalah contoh kasus untuk proses pencarian kata pada judul jurnal atau file dokumen, misalkan ada sebuah kasus pengunjung mencari judul, akan ada 2 teks tujuan yang akan dicari adalah “TMJ05-PANDUAN CEPAT” dan “BELAJAR CEPAT CODEIGNITER”, dan pattern yang akan dicari adalah “CEPAT”. Pengguna ingin mencari jurnal secara tepat, dan memilih algoritma untuk pencarian text tersebut.

Berikut Tahap Pencarian Kata :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTERN | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PATTER  N | C | E | P | A | T |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTER  N |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTER  N |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTERN |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTERN |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTERN |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTERN |  |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTER  N |  |  |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTERN |  |  |  |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N |  | C | E | P | A | T |
| PATTER  N |  |  |  |  |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N |  | C | E | P | A | T |
| PATTERN |  |  |  |  |  |  |  |  |  |  |  | C | E | P | A | T |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N |  | C | E | P | A | T |
| PATTER  N |  |  |  |  |  |  |  |  |  |  |  |  | C | E | P | A | T |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTER  N |  |  |  |  |  |  |  |  |  |  |  |  | C | E | P | A | T |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

Tabel 4.9 Berikut hasil pencocokkan string pada Teks pertama

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | T | M | J | 0 | 5 | - | P | A | N | D | U | A | N | C | E | P | A | T |
| PATTER  N | | | | | | | | | | | | | | C | E | P | A | T |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

Untuk tahap Pencarian teks pertama secara manual menggunakan Algoritma

*Knuth-Morris-Pratt* selesai setelah melakukan sebanyak 15 iterasi atau pergeseran.

Tahap Pencarian Teks Kedua :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R |  | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R |  | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R |  | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R | C | E | P | A | T | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  |  |  |  |  | C | E | P | A | T |  |  |  |  |  |  |  |  |  |  |  |  |

1. Tidak terjadi ketidakcocokan, maka pattern akan digeser sebanyak 1 karakter ke kanan

Tabel 4.10 Berikut hasil pencocokkan string pada Teks kedua

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TEKS | B | E | L | A | J | A | R | C | E | P | A T |  | C | O | D | E | I | G | N | I | T | E | R |
| PATTERN |  |  |  |  |  |  |  | C | E | P | A T |  |  |  |  |  |  |  |  |  |  |  |  |

1. Telah terjadi kecocokan, maka pattern tidak akan digeser lagi.

Untuk tahap Pencarian teks kedua secara manual menggunakan Algoritma

*Knuth-Morris-Pratt* selesai setelah melakukan sebanyak 9 iterasi atau pergeseran.

# BAB V PENUTUP

* 1. **Simpulan**

Setelah membangun sebuah layanan sistem *repository* berbasis *search engine* dapat diambil kesimpulan bahwa sistem *repository* sudah berjalan secara bersama. Sistem ini sudah berjalan melalui jaringan manapun yang memiliki internet. Untuk Jurnal atau *e-book* yang di *upload* oleh *user* akan langsung terdapat pada *storage user* tersebut. Pengujian dilakukan pada pencaharian file secara manual dan dibandingkan dengan pencaharian menggunakan algoritma *Knuth Morris Pratt*.

* + 1. Berdasarkan pengujian yang telah dilakukan sistem *repository* berbasis *search engine* dengan menerapkan algoritma *Knuth Morris Pratt* yang telah berjalan dan sudah mendapatkan hasil pencarian dalam sistem serta dapat mempermudah pengguna untuk menemukan kata pada judul TGA, Jurnal dan karya ilmiah yang berhubungan dengan kata kunci yang sedang menjadi pusat pencarian.
    2. Untuk pengujian secara manual hasil pencarian yang dilakukan oleh website berdasarkan judul TGA, jurnal dan karya ilmiah berhasil menampilkan kata yang ingin di cari di dalam database. Dan ketidak cocokan kata terjadi apabila data yang ingin dicari tidak ada dalam database. Pattern akan menyesuaikan dengan teks sebanyak nilai -n pada teks dilakukan pergeseran dari kiri ke kanan setiap karakter kata hingga terjadi kecocokan dari pattern dengan teks.

76

* + 1. Pencarian kata juga dapat menghitungkan waktu eksekusi dalam satuan detik, untuk setiap pengujian pada sistem dapat dianalisis bahwa setiap 1,1 *milisecond* untuk beberapa pengujian dalam setiap kumpulan kata yang dicari mulai dari pencarian satuan kata, pencarian kata yang sama dalam dua judul berbeda dan pencarian kata yang sama dalam satu judul.

# Saran

Penulis menyadari bahwa proyek dalam tugas akhir ini masih jauh dari sempurna. Oleh karena itu penulis memberikan beberapa saran agar sistem ini dapat dikembangkan sehinngga lebih bermanfaat.

* + 1. Layanan sistem *repository* berbasis *search engine* masih dapat dikembangkan menggunakan *framework,* misalnya Yii Framework, Codeigniter Framework, atau Laravel Framework sehingga keamanan sistem akan lebih terjaga dan aman.
    2. Server yang digunakan untuk sistem harus selalu *stand by* hidup, agar server dapat bekerja secara *real time* sesuai dengan waktu yang diatur.

# DAFTAR PUSTAKA

Aisa, S. (2014). *Implementasi Private Cloud Menggunakan Raspberry pi Untuk Pengaksesan Data pribadi*, 16.

Andre. (2014). [*https://w*](http://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-)*ww*[*.duniailkom*](http://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-)*.c*[*om/pengertian*](http://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-)*-*[*dan-fungsi-php-dalam-*](http://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-) *pemograman-web/*. Retrieved from [www.duniailkom.com.](http://www.duniailkom.com/)

Diartono, D. A. (2010). *Integrasi Sistem Presensi Finger Print dan Sistem SMS gateway untuk Monitoring Kehadiran Siswa*. Retrieved september 28, 2017, fr[om http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/114](http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/114)

Diartono, D. A. (2010). Integrasi Sistem Presensi Finger Print dan Sistem SMS Gateway untuk Monitoring Kehadiran Siswa. *Jurnal Teknologi Informasi DINAMIK*, 73.

Felix Andreas Susanto, J. A. (2015). *Implementasi Dashboard untuk Sistem Monitoring Bimbingan dan Konseling Siswa*. Retrieved 09 27, 2017, from <http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/4644>

Felix Andreas Sutanto, J. A. (2015). *Implementasi Dashboard untuk Sistem Monitoring Bimbingan dan Konseling Siswa*. Retrieved september 27, 2017, f[rom http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/4644](http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/4644)

Firmansyah. (2017). [*http://hadribonjay.it.student.pens.ac.id/*.](http://hadribonjay.it.student.pens.ac.id/) Retrieved from [http://hadribonjay.it.student.pens.ac.id/.](http://hadribonjay.it.student.pens.ac.id/)

Kandov, I. (2015). *Experimental model of low Power Nas and Cloud Drive Based on Raspberry Pi*, 3.

Mira Afrina, A. I. (2015). *Pengembangan Sistem Imformasi SMS Gateway Dalam Meningkatkan Layanan Komunikasi Sekitar Akademika Fakultas Ilmu Komputer Unsri*. Retrieved september 27, 2017, from <http://ejournal.unsri.ac.id/index.php/jsi/article/download/2265/1114>

Mulyadi, T. (2008). *Membangun Web Server Repository Ubuntu 10.04 di Laboratorium Sinyal dan Sistem Politeknik Negeri Lhokseumawe*, 10.

Putra, R. S. (2016). *Membangun Repositori Politeknik Negeri Lhokseumawe Menggunakan Cloud Computing*, 13.

Sa'diah, H. T. (2017). *Implementasi Algoritma Knuth Morris Pratt Pada Fungsi Pencarian Judul Tugas Akhir Repositiry*, 11.

Supriyanto, A. (2016). Perancangan Sistem Monitoring Pelanggaran Siswa Di SMAN 1 Gedangan Berbasis Web dan SMS Gateway. *Universitas Muhammadiyah Sidoarjo*, 12.

Widada, C. K. (2016). *Pasang Surut Pengelolaan Institutional Repository (IR) Perpustakaan Universitas Muhammadiyah Surakarta*, 12.

Widiasri, M. (2016). *Sistem repositori Tugas Akhir Mahasiswa dengan Fungsi Peringkat Okapi BM25*, 3.

Zaky, A. M. (2015). *Implementasi Algoritma Knuth Morris Pratt Pada Perancangan Game Hanacaraka*, 82.

# BIODATA PENULIS

Nama : Said Mahfud

NIM 1490343096

No. HP 082367804476

E-Mail [: smahfud6@gmail.com](mailto:smahfud6@gmail.com) Tempat/Tanggal Lahir : Cunda, 03 Maret 1997

Program Studi : Teknologi Rekayasa Komputer Jaringan Jurusan : Teknologi Informasi dan Komputer

Agama : Islam

Alamat : Jalan Darussalam No.58 Hagu Selatan, Lhokseumawe

Motto : Learn from the past, hope for tommorrow and live for today

Identitas Orang Tua :

* Ayah : Drs. Said Nazaruddin
* Pekerjaan : PNS
* Ibu : Cut Meutia
* Pekerjaan : Ibu Rumah Tangga

Riwayat Pendidikan :

* MIN Lhokseumawe
* SMP Negeri 2 Lhokseumawe
* SMA Negeri 1 Lhokseumawe
* Politeknik Negeri Lhokseumawe

Judul Tugas Akhir :

# MEMBANGUN SISTEM REPOSITORY BERBASIS SEARCH ENGINE MENGGUNAKAN ALGORITMA KNUTH MORRIS PRATT