**PEMBUATAN APLIKASI  
*DOCUMENT MANAGEMENT SYSTEM* (DMS)**

**KERJA PRAKTIK**

Diajukan sebagai Laporan Pelaksanaan Kerja Praktik  
Program Studi Informatika

Oleh :

**Asep Maulana Ismail**

**0617124001**



**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK UNIVERSITAS WIDYATAMA**

**BANDUNG**

**2018**

LEMBAR PENGESAHAN

**PEMBUATAN APLIKASI *DOCUMENT MANAGEMENT SYSTEM* (DMS)**

**KERJA PRAKTIK**

Diajukan sebagai Laporan Pelaksanaan Kerja Praktik  
Program Studi Informatika

Oleh :

**Asep Maulana Ismail**

**0617124001**

Telah disetujui dan disahkan di Bandung, Tanggal \_\_ / \_\_ / 2019

Menyetujui,

|  |  |
| --- | --- |
| Pembimbing Kampus,  Iwa Ovyawan Herlistiono, S.T., M.T. | Pembimbing Lapangan,  Titis Sutisna |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SURAT PERNYATAAN  Saya yang bertanda tangan dibawah ini :   |  |  |  | | --- | --- | --- | | Nama | : | Asep Maulana Ismail | | NPM | : | 0617124001 | | Tempat dan Tanggal Lahir | : | Bandung 15 Juni 1992 | | Alamat | : | Jl. Kebon Gedang V RT/RW 06/07 Kel. Maleer, Kec. Batununggal, 40274, Bandung |   Menyatakan bahwa telah melaksanakan kerja praktik di PT. Bee Solution Partners. Bila terbukti tidak demikian, saya bersedia menerima sanksi yang berlaku.  Bandung, \_\_/\_\_\_\_\_\_\_\_\_\_\_/ 2019  Asep Maulana Ismail |

ABSTRAK

Sebuah perusahaan sudah selayaknya menggunakan sistem informasi untuk mendukung segala aktivitas yang dilakukan. Karena sistem informasi membantu untuk mengontrol kinerja proses bisnis. Salah satu hal yang harus menjadi perhatian perusahaan adalah tentang pengelolaan dokumen perusahaan. Bila pengelolaan dokumen perusahaan kurang baik, maka perusahaan akan kesulitan dalam mencari dokumen yang dibutuhkan. Hal tersebut bisa berdampak buruk bagi perusahaan, salah satunya adalah perusahaan bisa kehilangan peluang kerjasama dengan pihak lain. Hal tersebut mungkin saja terjadi bila perusahaan tidak dapat memberikan dokumen yang dibutuhkan dengan tepat waktu dikarenakan proses pencarian yang terlalu lama atau tidak ditemukannya dokumen yang dimaksud.

*Document Management System* (DMS) adalah kontrol otomatis dari dokumen yang dicitrakan dan dokumen elektronik, gambar halaman, *spreadsheet*, pesan suara dan *e-mail*, dokumen pengelohan kata, dan dokumen lain melalui siklus hidupnya dalam organisasi, dari awal pembuatan hingga pengarsipan akhir atau perusakan.

Hasil dari penelitian ini adalah bahwa DMS yang dibangun dapat digunakan sebagai sistem pengelolaan dokumen. Hal tersebut dapat dilihat pada fitur-fitur dari DMS yaitu pencarian, pembacaan konten file, dan pengamanan dokumen yang sudah teruji dengan baik.

**Kata kunci : *Electronic Library, Document Management System, Scrum, Microservices, Search Engine***

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang maha pengasih lagi maha penyayang, segala puji dan syukur penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat-Nya sehingga dapat menuntaskan laporan kerja praktik yang berjudul **“PEMBUATAN APLIKASI *DOCUMENT MANAGEMENT SYSTEM*”**.

Penulis menyadari bahwa Laporan Kuliah Kerja Praktik ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun dari semua pihak sangat penulis harapkan guna perbaikan penulis pada masa yang akan datang.

Untuk itu dengan segala ketulusan, penulis menghaturkan terima kasih kepada:

1. **Kedua Orangtua,** yang selalu memberikan dorongan, kasih sayang, dan doa untuk kelancaran dalam penulisan laporan ini.
2. **Iwa Ovyawan Herlistiono, S.T., M.T.,** selaku Dosen pembimbing kampus yang telah memberikan bimbingan, terima kasih atas kritik dan saran Bapak dalam penyusunan laporan ini.
3. **Titis Sutisna**, selaku Pembimbing Lapangan di PT. Bee Solution Partners.
4. **Sriyani Violina, S.T., M.T.,** selaku Ketua Program Studi Informatika Fakultas Teknik Universitas Widyatama.
5. Pihak-pihak lain yang tidak bisa saya sebutkan satu persatu, terima kasih atas semua bantuan dan motivasinya.

Akhir kata semoga Allah SWT membalas segala kebaikan dari semua pihak yang telah membantu dalam penyusunan Laporan Kerja Praktik ini.

Bandung, Februari 2019

Asep Maulana Ismail

DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc366209)

[SURAT PERNYATAAN ii](#_Toc366210)

[ABSTRAK iii](#_Toc366211)

[KATA PENGANTAR iv](#_Toc366212)

[DAFTAR ISI v](#_Toc366213)

[DAFTAR GAMBAR viii](#_Toc366214)

[DAFTAR TABEL ix](#_Toc366215)

[BAB I PENDAHULUAN 1](#_Toc366216)

[1.1. Latar Belakang 1](#_Toc366217)

[1.2. Rumusan Masalah 2](#_Toc366218)

[1.3. Batasan Masalah 2](#_Toc366219)

[1.4. Tujuan 3](#_Toc366220)

[1.5. Metodologi Penelitian 3](#_Toc366221)

[1.6. Sistematika Penulisan 5](#_Toc366222)

[BAB II KEADAAN UMUM INSTANSI 7](#_Toc366223)

[2.1. Tinjauan Umum Perusahaan 7](#_Toc366224)

[2.2. Struktur Organisasi 8](#_Toc366225)

[BAB III LANDASAN TEORI 11](#_Toc366226)

[3.1. Tinjauan Pustaka 11](#_Toc366227)

[3.1.1. Document Management System 11](#_Toc366228)

[3.1.2. File Storage 12](#_Toc366229)

[3.1.3. Metodologi Scrum 12](#_Toc366230)

[3.1.4. Microservices 13](#_Toc366231)

[3.1.5. Black-box Testing 13](#_Toc366232)

[BAB IV ANALISIS KEBUTUHAN SISTEM 14](#_Toc366233)

[4.1. Proses Bisnis 14](#_Toc366234)

[4.1.1. Proses Login 14](#_Toc366235)

[4.1.2. Proses Upload Dokumen 15](#_Toc366236)

[4.1.3. Proses Pencarian Dokumen 16](#_Toc366237)

[4.1.4. Proses Preview Dokumen 17](#_Toc366238)

[4.1.5. Proses Download Dokumen 18](#_Toc366239)

[4.1.6. Proses Menghapus Dokumen 19](#_Toc366240)

[4.1.7. Proses Restore Dokumen 19](#_Toc366241)

[4.1.8. Proses Menghapus Dokumen Secara Permanen 20](#_Toc366242)

[4.2. Product Backlog 20](#_Toc366243)

[BAB V DOKUMENTASI SPRINT 23](#_Toc366244)

[5.1. Dokumentasi Sprint 1 23](#_Toc366245)

[5.1.1. Sprint Planning 23](#_Toc366246)

[5.1.2. Perancangan Sistem 23](#_Toc366247)

[5.1.3.1. Arsitektur Sistem 24](#_Toc366248)

[5.1.3.2. Desain Antarmuka Web App 25](#_Toc366249)

[5.1.3.3. Struktur Data User API 28](#_Toc366250)

[5.1.3.4. Struktur Data Search Engine 29](#_Toc366251)

[5.1.3. Implementasi 32](#_Toc366252)

[5.1.3.1. Antarmuka Login 32](#_Toc366253)

[5.1.3.2. Antarmuka Register 32](#_Toc366254)

[5.1.3.3. Antarmuka My Libray (Dasbor) 34](#_Toc366255)

[5.1.3.4. Antaramuka New Files 35](#_Toc366256)

[5.1.3.5. Antarmuka Bin 36](#_Toc366257)

[5.1.3.6. Antarmuka Pencarian 36](#_Toc366258)

[5.1.4. Pengujian 37](#_Toc366259)

[5.1.4.1. Pengujian Fitur Register 37](#_Toc366260)

[5.1.4.2. Pengujian Fitur Aktivasi 38](#_Toc366261)

[5.1.4.3. Pengujian Fitur Login 38](#_Toc366262)

[5.1.4.4. Pengujian Fitur Upload 39](#_Toc366263)

[5.1.4.5. Pengujian Fitur Search 39](#_Toc366264)

[5.1.4.6. Pengujian Fitur Preview 40](#_Toc366265)

[5.1.4.7. Pengujian Fitur Download 40](#_Toc366266)

[5.1.4.8. Pengujian Fitur Move To Bin 41](#_Toc366267)

[5.1.4.9. Pengujian Fitur Restore 41](#_Toc366268)

[5.1.4.10. Pengujian Fitur Delete 41](#_Toc366269)

[5.1.4.11. Pengujian Fitur Logout 42](#_Toc366270)

[5.1.4.12. Pengujian Keamanan 42](#_Toc366271)

[5.1.5. Fitur Gagal 43](#_Toc366272)

[5.1.6. Kesimpulan Pengujian 43](#_Toc366273)

[BAB VI PENUTUP 44](#_Toc366274)

[6.1. Kesimpulan 44](#_Toc366275)

[6.2. Saran 44](#_Toc366276)

[DAFTAR PUSTAKA 46](#_Toc366277)

DAFTAR GAMBAR

[Gambar 2.1 Logo PT. Bee Solution Partners[3] 8](#_Toc366296)

[Gambar 2.2. Struktur Organisasi PT. Bee Solution Partners[3] 9](#_Toc366297)

[Gambar 4.1. Proses Bisnis Keseluruhan 14](#_Toc366298)

[Gambar 4.2. Proses Bisnis *Login* 15](#_Toc366299)

[Gambar 4.3. Bisnis Proses *Upload* 16](#_Toc366300)

[Gambar 4.4. Bisnis Proses Pencarian Dokumen 17](#_Toc366301)

[Gambar 4.5. Bisnis Proses *Preview* Dokumen 18](#_Toc366302)

[Gambar 4.7. Bisnis Proses *Download* Dokumen 18](#_Toc366303)

[Gambar 4.6. Bisnis Proses Menghapus Dokumen 19](#_Toc366304)

[Gambar 4.8. Bisnis Proses *Restore* Dokumen 19](#_Toc366305)

[Gambar 4.9. Bisnis Proses Menghapus Dokumen Secara Permanen 20](#_Toc366306)

[Gambar 5.1. Arsitektur Umum Sistem 24](#_Toc366307)

[Gambar 5.2. Desain antarmuka *Login* 25](#_Toc366308)

[Gambar 5.3. Desain antarmuka *Register* 26](#_Toc366309)

[Gambar 5.4. Desain antarmuka *New Files* 26](#_Toc366310)

[Gambar 5.5. Desain antarmuka *My Library* 27](#_Toc366311)

[Gambar 5.6. Desain antarmuka *Bin* 27](#_Toc366312)

[Gambar 5.7. *Physical Data Model* Tabel *Users* 29](#_Toc366313)

[Gambar 5.8. Struktur data *Elastic search* 31](#_Toc366314)

[Gambar 5.9. Halaman *Login* 32](#_Toc366315)

[Gambar 5.10. Halaman *Register* 33](#_Toc366316)

[Gambar 5.11 Halaman *Resend* 33](#_Toc366317)

[Gambar 5.12. *Email Activation* 34](#_Toc366318)

[Gambar 5.13. Halaman *My Library* 35](#_Toc366319)

[Gambar 5.14. Halaman *New Files* 35](#_Toc366320)

[Gambar 5.15. Halaman *Bin* 36](#_Toc366321)

[Gambar 5.16. Fitur Pencarian 36](#_Toc366322)

DAFTAR TABEL

[Tabel 4.1. *Product Backlog Item* 21](#_Toc366278)

[Tabel 5.1. *Product Backlog Item Sprint 1* 23](#_Toc366279)

[Tabel 5.2. Fitur *User* API 28](#_Toc366280)

[Tabel 5.3. Fitur *Search Engine* 29](#_Toc366281)

[Tabel 5.4. Penjelasan atribut struktur data *Elastic search* 31](#_Toc366282)

[Tabel 5.5. Pengujian *Register* 37](#_Toc366283)

[Tabel 5.6. Pengujian Aktivasi 38](#_Toc366284)

[Tabel 5.7. Pengujian *Login* 38](#_Toc366285)

[Tabel 5.8. Pengujian *Upload* 39](#_Toc366286)

[Tabel 5.9. Pengujian *Search* 40](#_Toc366287)

[Tabel 5.10. Pengujian *Preview* 40](#_Toc366288)

[Tabel 5.11. Pengujian *Download* 41](#_Toc366289)

[Tabel 5.12. Pengujian *Move To Bin* 41](#_Toc366290)

[Tabel 5.13. Pengujian *Restore* 41](#_Toc366291)

[Tabel 5.14. Pengujian *Delete* 42](#_Toc366292)

[Tabel 5.16. Pengujian *Logout* 42](#_Toc366293)

[Tabel 5.17. Pengujian Keamanan 42](#_Toc366294)

[Tabel 5.18. Pengujian Kembali Fitur *Login* 43](#_Toc366295)

BAB I   
PENDAHULUAN

* 1. Latar Belakang

Seiring dengan kemajuan teknologi informasi, berbagai sistem informasi dibangun untuk membantu pekerjaan dalam mengelola *data* dan menyajikan *data* untuk memperoleh informasi yang bermanfaat bagi suatu instansi atau perusahaan. Melihat kenyataan tersebut, pengembangan sistem informasi menjadi salah satu ladang usaha.

Sebuah perusahaan sudah selayaknya menggunakan sistem informasi untuk mendukung segala aktivitas yang dilakukan. Karena sistem informasi membantu untuk mengontrol kinerja proses bisnis [1].

PT. Bee Solution Partners (BSP) merupakan perusahaan yang bergerak dibidang sistem informasi yang berperan sebagai konsultan. BSP didirikan oleh sejumlah tenaga pengajar pada suatu perguruan tinggi. BSP semula berperan sebagai penyalur tenaga kerja lulusan untuk dapat menempatkan lulusan pada tempat kerja yang sesuai dengan keahlian. Seiring dengan berkembangannya perusahaan, pembuatan produk perusahaan mulai dikembangkan.

Salah satu hal yang harus menjadi perhatian perusahaan adalah tentang pengelolaan dokumen perusahaan. Bila pengelolaan dokumen perusahaan kurang baik, maka perusahaan akan kesulitan dalam mencari dokumen yang dibutuhkan. Hal tersebut bisa berdampak buruk bagi perusahaan, salah satunya adalah perusahaan bisa kehilangan peluang kerjasama dengan pihak lain. Hal tersebut mungkin saja terjadi bila perusahaan tidak dapat memberikan dokumen yang dibutuhkan dengan tepat waktu dikarenakan proses pencarian yang terlalu lama atau tidak ditemukannya dokumen yang dimaksud.

*Document Management System* (DMS) adalah sistem komputer yang dapat mengidentifikasi penyimpanan, mengakses, melacak, dan menampilkan informasi dalam format elektronik untuk membantu dalam menentukan keputusan.

Digitasi dokumen perusahaan dan penggunaan DMS bisa menjadi salah satu solusi untuk mengatasi permasalahan tersebut. Perusahaan dapat memindai dokumen dari seluruh kegiatan perusahaan, lalu memasukannya kedalam DMS agar lebih mudah dalam melakukan proses pencarian.

Scrum adalah metodologi pengembangan perangkat lunak yang akan digunakan dalam membangun DMS ini. Scrum adalah kerangka kerja untuk manajemen proyek yang menekankan kepada kerja tim, akuntabilitas, dan iteratif untuk menuju tujuan yang jelas[6]. Dan untuk pengembangan DMS saat ini dibatasi hanya pada Sprint 1.

* 1. Rumusan Masalah

Berdasarkan latar belakang tersebut, pembuatan DMS diperlukan karena beberapa permasalahan, diantaranya:

1. Bagaimana sistem mampu menampilkan dokumen yang relevan dengan kata kunci pencarian?
2. Bagaimana sistem mampu membaca konten dari *file* dokumen?
3. Bagaimana cara menjaga *file* yang sudah diunggah pengguna agar tidak dapat diakses oleh pengguna lain?
   1. Batasan Masalah

Sehubungan dengan keterbatasan yang dimiliki, maka dalam pembuatan DMS ini akan dibatasi oleh beberapa hal, diantaranya::

1. *File* yang didukung pada sistem DMS hanya *file* bertipe PDF dengan ukuran maksimal 10 MB.
2. Sistem mampu melakukan ekstraksi konten *file* PDF.
3. Pencarian hanya akan melibatkan nama *file* dan konten yang berhasil diekstrak dari file PDF.
4. Sistem yang dibangun hanya akan memiliki fitur yang mengacu kepada *Backlog Item* pada *Sprint 1*.
5. Fitur OCR tidak diimplementasi pada *Sprint 1*.
   1. Tujuan

Tujuan dari kerja praktek ini diantaranya:

1. Membuat sistem yang mampu menampilkan dokumen yang relevan dengan kata kunci pencarian.
2. Membuat sistem yang mampu membaca konten dari *file* dokumen.
3. Membuat sistem yang mampu mengamankan *file* pengguna dari akses pengguna lain.
   1. Metodologi Penelitian

Dalam kerja praktek ini, berikut metode penelitian yang digunakan.

1. **Metode Pengumpulan Data**

Kerja praktik ini dilakukan sebagai untuk menambah pengetahuan dan pengalaman bekerja di perusahaan. Adapun kegiatan yang dilakukan yaitu :

1. Studi Literatur

Mempelajari dokumentasi sistem, *framework,* bahasa, dan *database* yang digunakan.

1. Wawancara

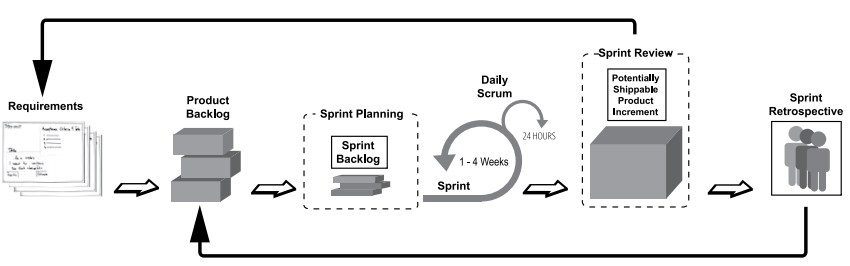
Melakukan wawancara secara langsung dengan analis sistem terkait keperluan dan spesifikasi yang dibutuhkan. Wawancara merupakan salah satu kegiatan penting dalam memahami bisnis proses DMS yang akan dibangun.

1. Studi Pustaka

Mempelajari buku-buku serta berbagai referensi lain yang ada kaitannya dalam perancangan sistem dan penyusunan laporan kerja praktik ini.

1. **Metode Pembangunan Perangkat Lunak**

Model pengembangan sistem informasi yang digunakan adalah model pengembangan *Scrum*  yang terdiri dari beberapa tahapan :



Gambar 1.5.2.1. Model pengembangan *Scrum* [2]

#### Product Backlog

*Product Backlog* adalah artefak yang sebenarnya dari *framework* scrum. *Product backlog* ini berperan sebagai daftar pekerjaan utama yang harus dilakukan dari sebuah proyek [2].

#### Sprint Planning

*Sprint Planning* adalah fase yang dilakukan ketika *system requirements* telah terdefinisi oleh tim scrum, lalu tim *development* membagi *system requirements* tersebut menjadi daftar pekerjaan individu [2].

#### Sprint Backlog

*Sprint Backlog* adalah hasil yang didapatkan dari sesi *sprint planning*. *Sprint backlog* berisi tentang urutan daftar *requirements* dan tugas-tugas yang harus dilakukan untuk dapat mencapai tujuan dari *sprint* [2]

#### Sprint

*Sprint* adalah sesi dimana kita melakukan pengembangan berdasarkan *sprint backlog* yang telah didefinisikan sebelumnya. Setiap *sprint* diharapkan menghasilkan sebuah produk yang bisa digunakan (*most viable product*) [2].

#### Sprint Review

*Sprint review* adalah fase dimana *product owner* memberikan *feedback* terhadap produk yang sedang dibangun dengan meyakinkan bahwa produk yang sedang dibangun sesuai dengan keinginan *product owner* [2].

#### Sprint Retrospective

*Sprint retrospective* adalah fase untuk memberikan kesempatan bagi tim scrum untuk menilai apa saja yang berjalan baik dan apa saja yang perlu ditingkatkan dari *sprint* yang sudah dilakukan [2].

* 1. Sistematika Penulisan

Sistematika penulisan laporan kerja praktik ini adalah sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, metode penelitian dan sistematika penulisan.

**BAB II KEADAAN UMUM INSTANSI**

Bab ini berisi tentang gambaran perusahaan tempat kerja praktik. Terdiri dari tinjauan umum perusahaan, struktur organisasi, dan deskripsi kerja.

**BAB III LANDASAN TEORI**

Bab ini berisi tentang teori - teori singkat yang menunjang dalam laporan kerja praktik.

**BAB IV ANALISIS KEBUTUHAN SISTEM**

Bab ini membahas mengenai uraian analisis kebutuhan sistem dengan tujuan untuk mendapatkan requirement dari sistem yang akan dibangun. *Output* yang dihasilkan dari bab ini adalah berupa *product backlog*.

**BAB V DOKUMENTASI SPRINT**

Bab ini membahas mengenai implementasi mulai dari perancangan, pengembangan, dan pengujian  sistem dari setiap sprint yang dilakukan.

**BAB VI PENUTUP**

Bab ini berisi mengenai kesimpulan yang didapat dari analisis mengenai keterkaitan dengan tujuan pembuatan sistem serta saran – saran yang berkaitan dengan penggunaan sistem atau pengembangan sistem di masa akan datang yang diperlukan dalam mendapatkan hasil yang lebih baik.

BAB II  
KEADAAN UMUM INSTANSI

1. Tinjauan Umum Perusahaan

PT. Bee Solution Partners (BSP) didirikan pada tanggal 18 Juni 2011. Selama masa perkembangannya, BSP mampu bertahan dan menjadi perusahaan yang dipercaya oleh beberapa perusahaan besar baik di Indonesia maupun luar negeri [3].

BSP adalah sebuah perusahaan yang bergerak di bidang Teknologi Informasi dan berada di Kota Bandung. Sebagai perusahaan muda yang dinamis, BSP berkeinginan besar untuk dapat memberikan solusi layanan teknologi terbaik, inovatif, kreatif dan unik. Dukungan dari sumber daya manusia yang kompeten dan profesional, komunitas kampus, juga dari mitra-mitra industri yang strategis akan menciptakan sebuah sinergi yang strategis, efektif dan dahsyat [3].

BSP sejak lahirnya merupakan perusahaan berbasis java dan seiring berkembangnya waktu, BSP memperluas lingkup solusinya ke beberapa bahasa dan *framework* lain. Saat ini diperkuat oleh lebih dari 40 orang tenaga profesional. Pada awalnya BSP terfokus pada perusahaan *outsourcing*, Namun seiring dengan perkembangan waktu, BSP mulai merambah beberapa bidang bisnis yang diharapkan dapat menjadi solusi perkembangan Teknologi Informasi di Indonesia. Mulai tahun 2012, BSP telah berkembang dan fokus pada penyedia Jasa *Bootcamp*, *Consultant* dan *Offshore*. Berikut penjelasan mengenai jasa-jasa yang diberikan oleh BSP [3].

1. **Bootcamp**

*Bootcamp* merupakan jasa yang diberikan PT. Bee Solution Partners kepada konsumen yang ingin melakukan pelatihan menggunakan *tools* atau *software*. Contohnya pelatihan untuk mempelajari bahasa pemrograman Java, PHP, dll [3].

1. **Consultant**

*Consultant* merupakan jasa yang diberikan PT. Bee Solution Partners kepada konsumen yang ingin mengembangkan perangkat lunak atau ingin mengetahui bagaimana menggunakan perangkat lunak yang relevan dengan tujuan perusahaan[3].

1. **Offshore**

*Offshore* merupakan jasa yang diberikan PT. Bee Solution Partners kepada konsumen yang ingin membeli Perangkat Lunak maupun mengembangkan sistem yang sesuai dengan kebutuhan konsumen[3].

Adapun logo PT. Bee Solution Partners dapat dilihat pada gambar berikut :

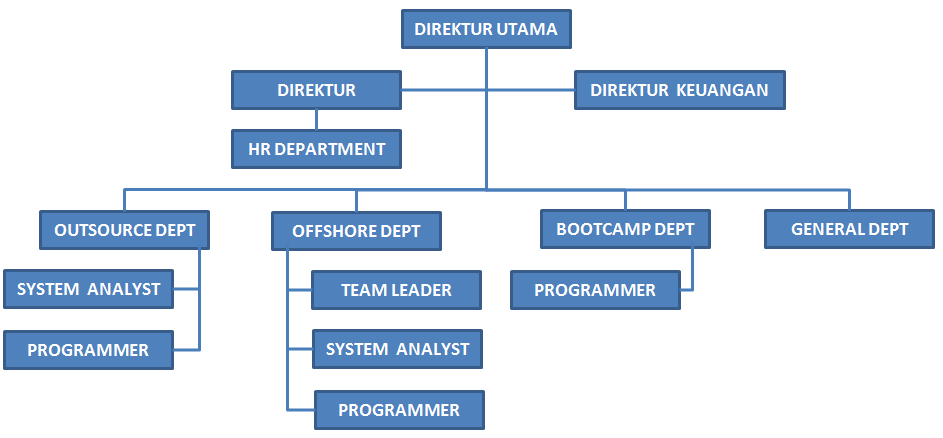


Gambar 2.1 Logo PT. Bee Solution Partners[3]

Logo perusahaan PT. Bee Solution Partners berasal dari bentuk sarang lebah yang berarti tempat lebah mengunpukan madu. Lebah adalah serangga yang kecil namun bekerja keras mengumpulkan madu yang memilik manfaat yang banyak. Maka dari itu sarang lebah dianalogikan sebagai perusahaan yang merupakan tempat bekerja keras untuk dapat menebar manfaat.

1. Struktur Organisasi

Struktur organisasi PT. Bee Solution Partners dapat dilihat pada gambar berikut :



Gambar 2.2. Struktur Organisasi PT. Bee Solution Partners[3]

Berikut merupakan deskripsi kerja dari masing-masing bagian organisasi :

1. Direktur Utama

Direktur utama berperan untuk memutuskan dan menentukan peraturan dan kebijakan tinggi perusahaan juga bertanggung jawab dalam memimpin dan menjalankan perusahaan.

1. Direktur

Direktur berperan sebagai pihak yang menandatangai kontrak proyek dan bertanggung jawab terhadap proyek – proyek yang dikerjakan dalam perusahaan.

1. Direktur Keuangan

Direktur keuangan berperan untuk mengatur anggaran kas proyek yang dikerjakan dalam perusahaan dan anggaran kebutuhan operasional perusahaan.

1. *Human Resource (HR) Department*

*Human Resource Deptartement* berperan untuk pengajian, perekrutan dan perencanaan pengembangan kualitas karyawan.

1. *Outsource Dept*

*Outsource Departement* berperan untuk mengatur administrasi dan bertanggung jawab terhadap karyawan yang sedang di-*outsource* di perusahaan *client*.

1. *Offshore Dept*

*Offshore Departement* beperan untuk mengatur administrasi, mengawasi dan mengkontrol pengerjaan proyek dari perusahaan *client* yang dikerjakan di perusahaan.

1. *Bootcamp Dept*

*Bootcamp Departement* berperan untuk melakukan pelatihan – pelatihan untuk pengembangan diri karyawan.

1. *General Dept*

*General Departement* berperan untuk mengatur admistrasi perusahaan terkait pelaporan pajak, penggajian dan kebutuhan operasional sehari-hari.

1. *System Analyst*

Pekerjaan *system* *analyst* dalam pembuatan program terbatas pada pemecahan masalah secara garis besar. Pekerjaan *system analyst* melibatkan hubungan banyak orang, tidak terbatas pada sesama *system analyst*, *programmer* tetapi juga pemakai sistem dan manajer. *System analyst* termasuk orang yang melakukan uji coba terhadap program dan orang yang melakukan pembuatan dokumentasi teknis.

1. *Programmer*

Tanggung jawab *programmer* terbatas pada pembuatan program komputer. Pekerjaan *programmer* sifatnya teknis dalam pembuatan instruksi - instruksi program dan tidak menyangkut hubungan dengan banyak orang, terbatas hanya pada sesama *programmer* dan *system analyst*. Termasuk pelaksanaan perbaikan *bug* (*bug fixing)*.

1. *Team Leader*

*Team Leader* bertugas dalam mengawasi pekerjaan team yang terdiri dari *Programmer* dan *System Analyst*. Bertugas untuk mengatur jadwal dan *jobdesk* dari tiap-tiap anggota timnya.

BAB III  
LANDASAN TEORI

1. Tinjauan Pustaka
2. Document Management System

*Document Management System* (DMS) adalah kontrol otomatis dari dokumen yang dicitrakan dan dokumen elektronik, gambar halaman, *spreadsheet*, pesan suara dan *e-mail*, dokumen pengelohan kata, dan dokumen lain melalui siklus hidupnya dalam organisasi, dari awal pembuatan hingga pengarsipan akhir atau perusakan[4].

DMS terdiri *hardware* dan *software* yang digunakan untuk mengelola arsip dokumen elektronik dan untuk mengkonversi dokumen kertas menjadi dokumen elektronik, dan juga untuk melakukan *indexing* dan menyimpannya secara terorganisir. Sebagai contoh, sebuah perusahaan mungkin diharuskan oleh hukum untuk menyimpan dokumen keuangan setidaknya selama tujuh tahun, sedangkan pesan *email* yang berkaitan dengan promosi pemasaran akan disimpan selama satu tahun sebelum dibuang. DMS memiliki kemampuan pencarian sehingga dapat diidentifikasi dan diakses seperti *data* dalam *database*[4].

Departemen atau perusahaan yang karyawannya menghabiskan sebagian besar hari untuk pengarsipan atau mengambil dokumen atau menyimpan catatan kertas dapat mengurangi biaya secara signifikan dengan DMS. Sistem ini meminimalkan inefisiensi dan frustasi yang terkait dengan pengelolaan dokumen kertas. Namun, secara signifikan, mereka tidak melakukannya dengan menghilangkan kertas sama sekali seperti yang telah diprediksi.

DMS dapat membantu bisnis menjadi lebih efisien dan produktif dengan:

1. Memungkinkan perusahaan untuk mengakses dan menggunakan konten yang terdapat dalam dokumen
2. Memotong biaya tenaga kerja dengan mengotomatisasi proses bisnis
3. Mengurangi waktu dan upaya yang diperlukan untuk mencari informasi yang dibutuhkan bisnis
4. Meningkatkan keamanan konten, sehingga mengurangi risiko pencurian kekayaan intelektual.
5. Meminimalkan biaya yang terkait dengan percetakan, penyimpanan, dan pencarian konten [4].
6. File Storage

*File Storage* atau disebut juga dengan *file-level* atau *file-based storage*, adalah penyimpanan data didalam struktur hirarki. Data disimpan dalam bentuk *file* dan *folder*, dan ditampilkan dalam format yang sama ketika menyimpan dan juga ketika menampilkan[5].

Data dapat diakses menggunakan protokol *Network File System* (NFS) pada Unix atau Linux, atau menggunakan protokol *Server Message Block* (SMB) untuk Microsoft Windows[5].

1. Metodologi Scrum

Scrum adalah kerangka kerja untuk manajemen proyek yang menekankan kepada kerja tim, akuntabilitas, dan iteratif untuk menuju tujuan yang jelas. Kerangka kerja dimulai dengan premis sederhana: Mulailah dengan apa yang bisa dilihat atau diketahui. Setelah itu, lacak progresnya dan sesuaikan seperlunya. Tiga pilar Scrum adalah transparansi, inspeksi, dan adaptasi[6].

Ketika memasuki tahap pengembangan produk, tim Scrum akan terbagi menjadi beberapa peran yaitu *product owner*, Scrum *master*, dan Scrum *development team*[6]*.*

*Product owner* adalah anggota tim yang berperan sebagai jembatan antara tim development dan pengguna. *Product owner* bertanggung jawab dalam memastikan bahwa spesifikasi dari produk yang akan dibangun telah komunikasikan dan disepakati bersama dengan pengguna[6]*.*

Scrum *master* adalah anggota tim yang berperan sebagai *fasilitator*. Seorang Scrum *master* bertanggung jawab dalam keberlangsungan *project* dalam mencapai tujuan[6].

Scrum *development* *team* adalah sekelompok anggota yang bekerjasama dalam membuat dan menguji setiap produk dirilis hingga produk akhir selesai dibangun[6].

1. Microservices

*Microservices* adalah teknik pengembangan perangkat lunak yang membagi sistem yang akan dibangun menjadi beberapa *services* yang berdiri masing-masing. Manfaat dari arsitektur microservices ini yaitu membuat sistem yang akan dibangun menjadi lebih mudah dipahami, dikembangkan, dan diuji [7].

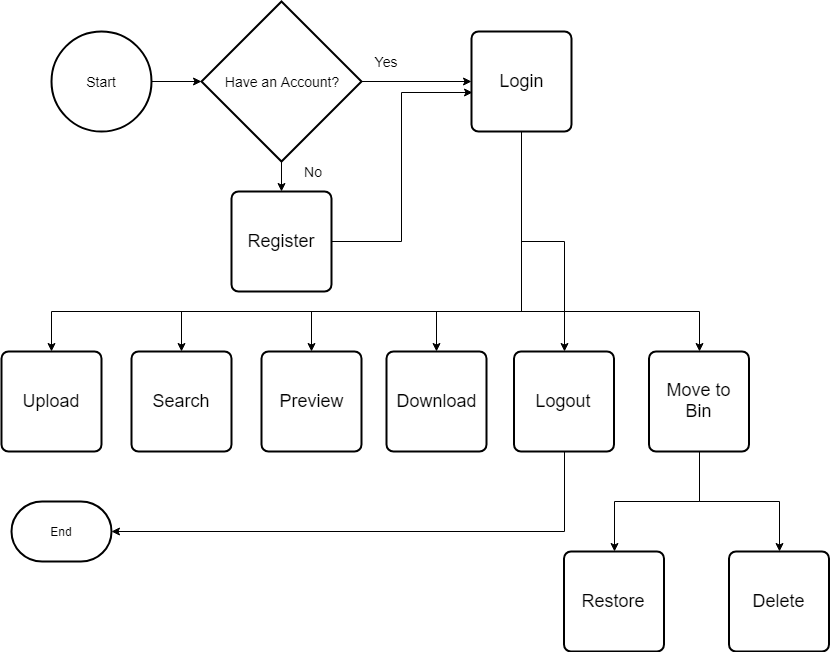
1. Black-box Testing

*Black-box Testing* adalah sebuat metodologi pengujian yang menguji setiap fitur dengan menghiraukan bagaimana fitur itu dibuat. *Black-box Testing* mengirimkan *input* terhadap suatu fitur, lalu membandingkan antara hasil yang diharapkan dengan hasil yang sebenarnya. Bila hasil yang sebenarnya sesuai dengan hasil yang diharapkan, maka fitur tersebut dapat dikatakan berhasil [8].

BAB IV  
ANALISIS KEBUTUHAN SISTEM

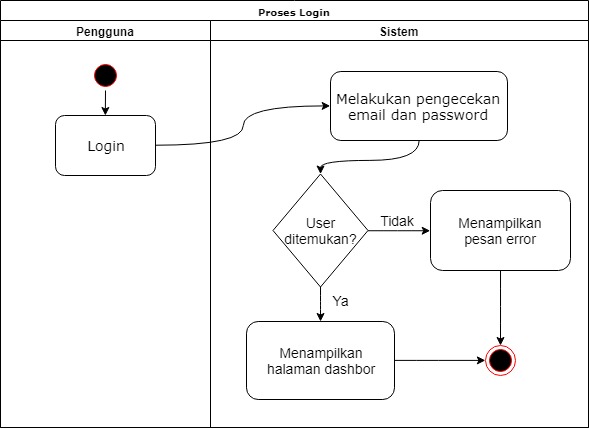
1. Proses Bisnis

Langkah yang pertama dilakukan untuk menganalisis kebutuhan sistem adalah dengan melihat bisnis proses yang terjadi. Berdasarkan hasil diskusi dengan pengguna, gambaran umum proses bisnis dari sistem ini adalah sebagai berikut:

  
Gambar 4.1. Proses Bisnis Keseluruhan

* 1. Proses Login

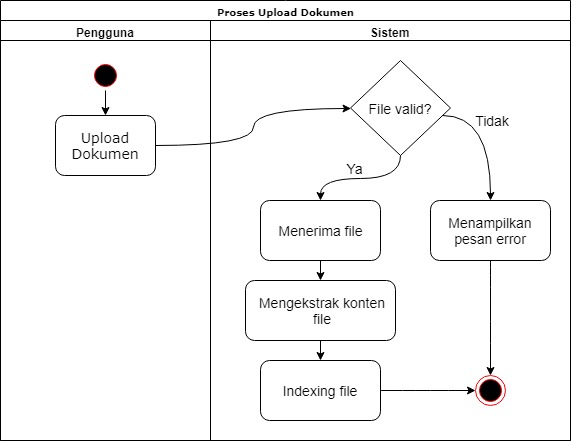
Proses ini adalah poses dimana sistem melakukan pengecekan apakah pengguna dapat menggunakan sistem atau tidak berdasarkan *email* dan *password* yang diisi oleh pengguna. Bila *email* dan *password* sesuai dengan yang ada pada *userlist*, maka sistem akan mengarahkan pengguna ke halaman dasbor. Bila tidak sesuai, maka sistem akan menampilkan pesan *error*.



Gambar 4.2. Proses Bisnis *Login*

* 1. Proses Upload Dokumen

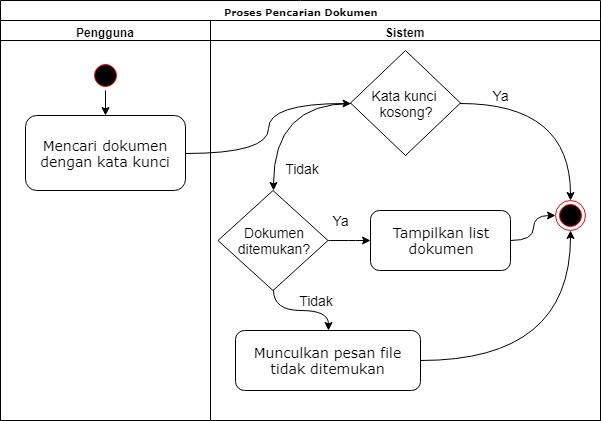
Proses ini adalah proses dimana sistem menerima *file* yang diunggah oleh pengguna. Sistem akan melakukan pengecekan terhadap setiap *file* yang diunggah, apakah format *file* sesuai dan ukuran *file* dibawah dari ukuran maksimum yang diperbolehkan untuk diunggah. Bila *file* dinyatakan *valid*, maka *file* selanjutnya akan disimpan dan dilakukan proses *content extraction* sehingga konten dari *file* tersebut dapat di-*index* oleh sistem.



Gambar 4.3. Bisnis Proses *Upload*

* 1. Proses Pencarian Dokumen

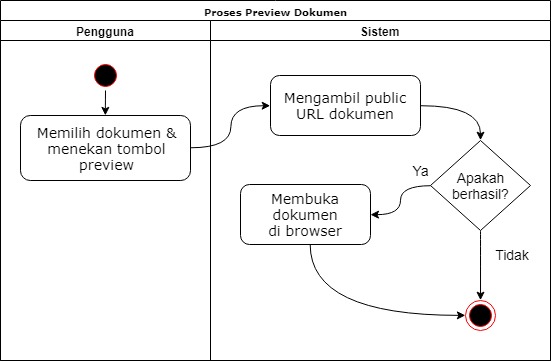
Proses ini adalah proses dimana sistem melakukan pencarian terhadap dokumen-dokumen yang telah diunggah berdasarkan kata kunci yang digunakan oleh pengguna. Bila dokumen yang sesuai dengan kata kunci ditemukan, maka sistem akan memunculkan daftar *file* tersebut, namun bila tidak ditemukan, sistem akan menampilkan pesan bahwa tidak ada *file* yang sesuai dengan kata kunci.



Gambar 4.4. Bisnis Proses Pencarian Dokumen

* 1. Proses Preview Dokumen

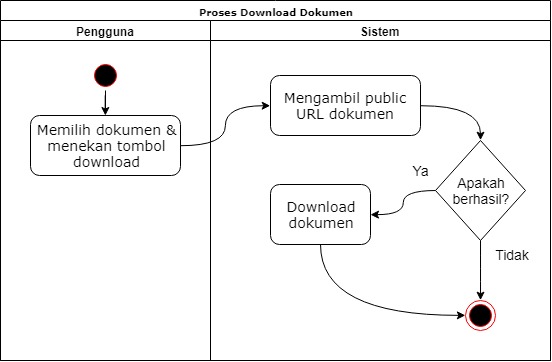
Proses ini adalah proses dimana sistem memberikan pratayang/*preview* dari dokumen. *Preview* ini mengizinkan pengguna untuk melihat konten *file* secara langsung tanpa mendownloadnya.



Gambar 4.5. Bisnis Proses *Preview* Dokumen

* 1. Proses Download Dokumen

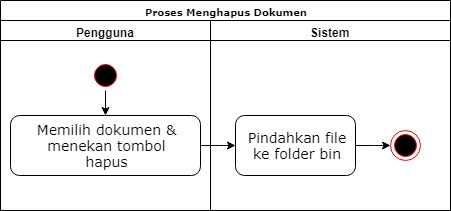
Proses ini adalah proses dimana sistem memberikan izin kepada pengguna untuk mengunduh suatu *file*.



Gambar 4.7. Bisnis Proses *Download* Dokumen

* + 1. Proses Menghapus Dokumen

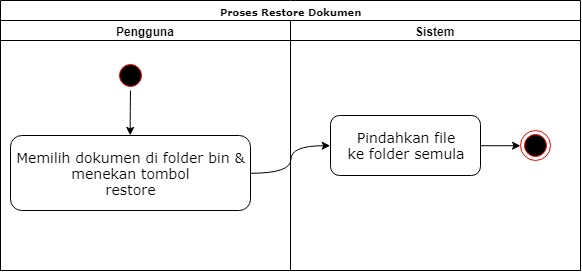
Proses ini adalah proses dimana sistem melakukan penghapusan terhadap suatu *file*. Untuk menghindari penghapusan *file* secara tidak sengaja, maka pada proses ini sistem tidak benar-benar menghapus *file*, akan tetapi sistem memindahkan *file* ke dalam *folder* sampah/*bin*.



Gambar 4.6. Bisnis Proses Menghapus Dokumen

* + 1. Proses Restore Dokumen

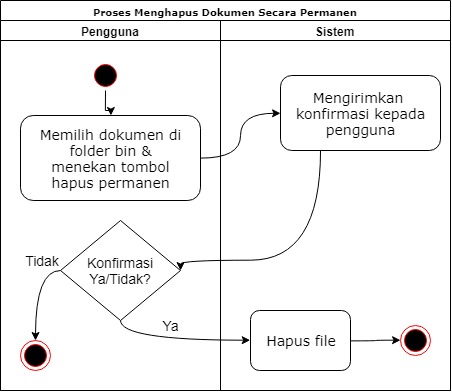
Proses ini adalah proses dimana sistem mengembalikan file yang ada di dalam *folder bin* ke *folder* sebelumnya.



Gambar 4.8. Bisnis Proses *Restore* Dokumen

* + 1. Proses Menghapus Dokumen Secara Permanen

Proses ini adalah proses dimana sistem melakukan penghapusan *file* secara permanen. Semua *file* yang sudah dihapus melalui proses ini tidak akan bisa dikembalikan lagi.

.

Gambar 4.9. Bisnis Proses Menghapus Dokumen Secara Permanen

* 1. Product Backlog

Berdasarkan proses bisnis yang sudah terdefinisi sebelumnya, terdapat beberapa fitur yang harus dimiliki oleh DMS untuk mendukung proses bisnis tersebut.

Pertama, sistem harus mampu dijadikan sebagai sistem pengelolaan *file* yang baik. Itu berarti fitur utama dari DMS yang akan diambil untuk menyelesaikan permasalahan ini yaitu: *Document Upload* dan *Document Storage*.

Kedua, sistem harus mampu melakukan pencarian terhadap dokumen dan menghasilkan dokumen-dokumen yang relevan dengan kata kunci yang digunakan. Dan terkadang informasi yang dicari oleh pengguna tidak terdapat pada nama *file*, sehingga sistem ini juga harus mampu melakukan pencarian ke dalam konten dari *file* yang telah diunggah. Untuk menyelesaikan permasalahan ini, sistem harus memiliki fitur: *Document Indexing*, *Full-text Search*.

Ketiga, terdapat dua jenis *file* pdf yang akan diunggah oleh pengguna, yaitu dokumen murni yang merupakan hasil dari pembuatan manual, dan dokumen yang merupakan hasil pemindaian (*scan*) dari dokumen fisik. Agar sistem ini mampu mengekstrak konten dokumen dari *file* yang merupakan hasil pemindaian, maka sistem ini pun harus memiliki fitur *Optical Character Recognition* (OCR).

Keempat, untuk menjaga dokumen dari hal-hal yang tidak diharapkan seperti pencurian dokumen, maka sistem ini harus memiliki fitur *User Authentication* sehingga setiap pengguna hanya dapat melakukan akses terhadap dokumen milikinya sendiri.

Kelima, untuk memudahkan pengguna dalam mengakses dokumen yang telah diunggah, baik untuk melihat isi dokumen dan juga mengunduh dokumen, maka sistem harus memiliki fitur *Document Download* dan *Document Preview*.

Berdasarkan pembahasan diatas, maka dapat dihasilkan *Product Backlog* dari *Document Management System* (DMS) yang akan dibuat, diantaranya:

Tabel 4.1. *Product Backlog Item*

| **No.** | ***Product Backlog Item*** |
| --- | --- |
| 1 | *Document Upload* |
| 2 | *Document Storage* |
| 3 | *Full-Text Search* |
| 4 | *Optical Character Recognition* |
| 5 | *User Authentication* |
| 6 | *Document Download* |
| 7 | *Document Preview* |

BAB V  
DOKUMENTASI SPRINT

1. Dokumentasi Sprint 1
2. Sprint Planning

Berdasarkan *Product Backlog* yang telah didapatkan sebelumnya, pada *sprint* 1 ini product yang dihasilkan harus memiliki fitur dibawah ini:

Tabel 5.1. *Product Backlog Item Sprint 1*

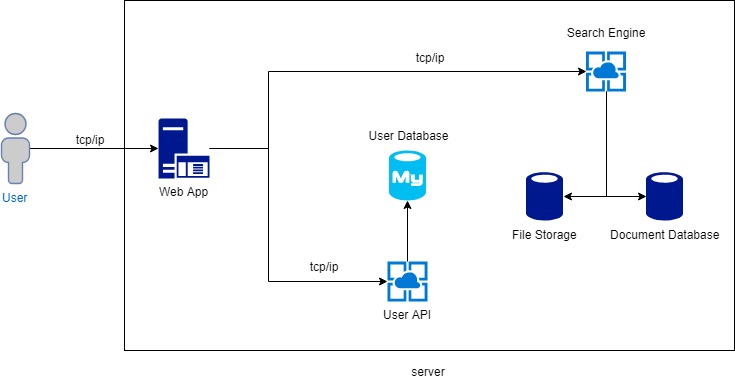
|  |  |
| --- | --- |
| ***Sprint*** | ***Product Backlog Item*** |
| *Sprint 1* | *Document Upload* |
| *Document Storage* |
| *Full-Text Search* |
| *User Authentication* |
| *Document Download* |
| *Document Preview* |

1. Perancangan Sistem

Perancangan sistem dimaksudkan untuk membantu menyelesaikan masalah pada sistem yang sedang berjalan saat ini sehingga dapat menjadi lebih baik dengan adanya sistem terkomputerisasi. Perancangan sistem dilakukan dalam setiap *sprint*.

1. Arsitektur Sistem

Perancangan pertama kali difokuskan bagaimana arsitektur sistem kita akan dibangun. Arsitektur yang akan digunakan yaitu *Microservices*, dan bila dilihat secara keseluruhan akan seperti gambar berikut:



Gambar 5.1. Arsitektur Umum Sistem

Berdasarkan gambar diatas, sistem memiliki dua *microservices* yaitu *User API* dan *Search Engine*, dimana masing-masing *microservices* tersebut berdiri sendiri dan saling berkomunikasi melalui HTTP API. Dan kedua *microservices* tersebut akan dihubungkan oleh sebuah *orchestrator* berupa *Web Application*.

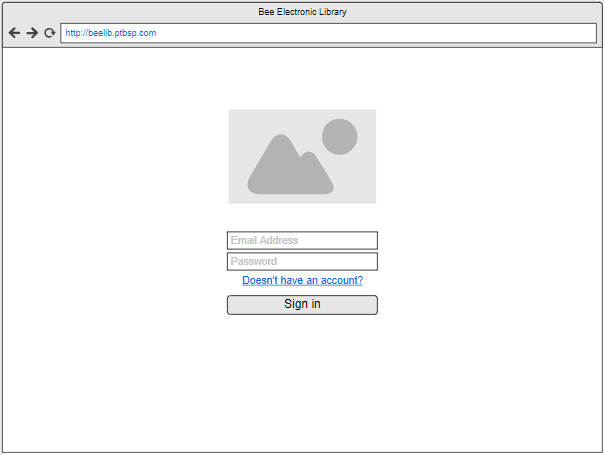
*User* API ini merupakan sebuah API yang menangani hal-hal yang berkaitan dengan pengguna. Dan *microservice* ini akan menggunakan MySQL sebagai basis datanya. Sedangkan *Search Engine* adalah *microservice* utama yang ada pada sistem ini, karena mayoritas fitur pada sistem ini terdapat pada *Search Engine*. *Search Engine* ini merupakan sebuah API yang menangani hal-hal yang berkaitan dengan dokumen, dimana API ini akan menggunakan Minio sebagai media penyimpanan *file*, dan *Elastic Search* sebagai media penyimpanan metadata dari *file* tersebut.

*Web Application* merupakan aplikasi yang menghubungkan antara dua *microservices* tersebut sekaligus sebagai antarmuka pengguna dalam menggunakan sistem ini. Di dalam arsitektur *microservices*, *Web Application* biasa disebut sebagai *Orchestrator*.

1. Desain Antarmuka Web App

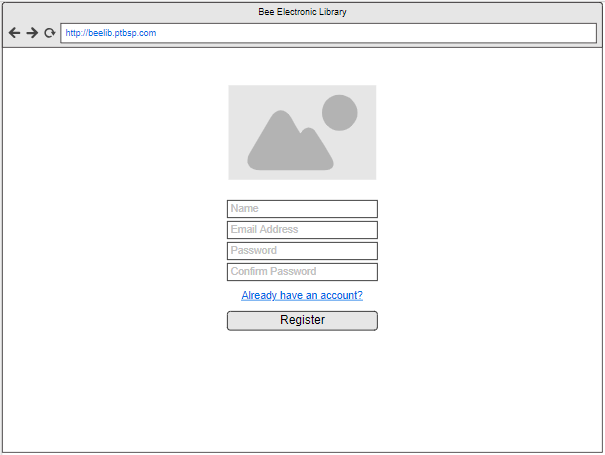
*Web Application*  dibangun diatas *framework* Laravel dengan struktur MVC (*Model-View-Controller*). Namun karena *Web App* ini hanya digunakan sebagai *orchestrator* dari sistem, maka *Model* dalam *framework* Laravel ini tidak digunakan karena tidak memiliki akses langsung terhadap *database*. Untuk pengambilan data, *Web App* ini akan meminta kepada *User API* atau *Search Engine* melalui HTTP API. Sedangkan untuk antarmuka pengguna, *Web App* ini akan memanfaatkan *framework* VueJS.

Halaman *login* adalah halaman yang harus digunakan pengguna sebelum dapat masuk kedalam aplikasi. Berikut adalah perancangan antarmuka dari fitur *login*.



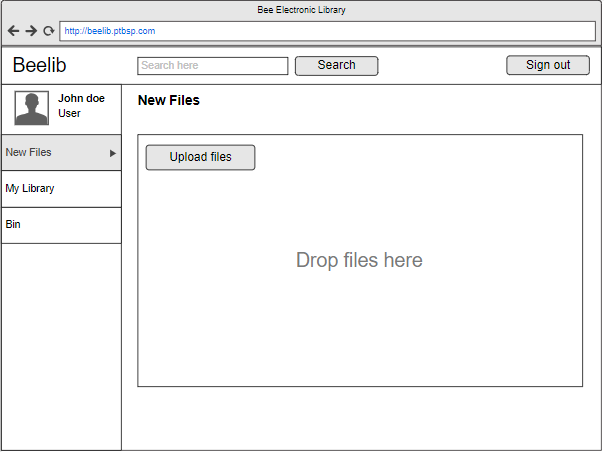
Gambar 5.2. Desain antarmuka *Login*

Ketika pengguna sama sekali belum memiliki akun, maka pengguna harus melakukan pendaftaran terlebih dahulu melalui halaman registrasi. Berikut adalah perancangan antarmuka dari fitur registrasi.



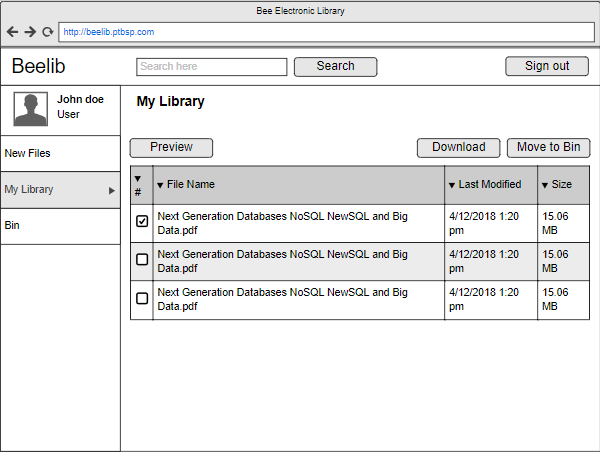
Gambar 5.3. Desain antarmuka *Register*

Ketika pengguna masuk ke dalam aplikasi, maka pengguna dapat mengunggah dokumen melalui halamana *new files*. Berikut adalah perancangan antarmuka dari fitur *new files*.



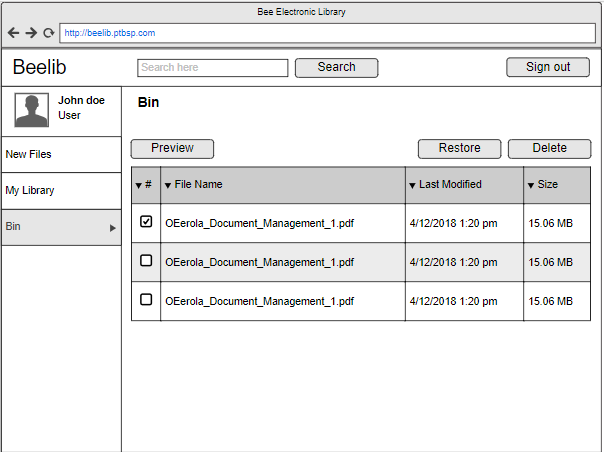
Gambar 5.4. Desain antarmuka *New Files*

Untuk melihat daftar *file* yang telah diunggah, pengguna dapat menggunakan fitur *my library*. Berikut adalah perancangan antarmuka dari fitur *my library*.



Gambar 5.5. Desain antarmuka *My Library*

Untuk meminimalisir resiko kesalahan dalam menghapus *file*, maka setiap *file* yang dihapus oleh pengguna pertama kali akan masuk ke dalam *folder bin*. Berikut adalah perancangan antarmuka dari fitur *bin*.



Gambar 5.6. Desain antarmuka *Bin*

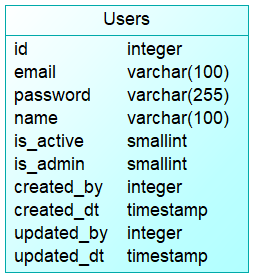
1. Struktur Data User API

*User* API memiliki beberapa fitur diantaranya:

Tabel 5.2. Fitur *User* API

|  |  |  |
| --- | --- | --- |
| **No.** | **Fitur** | **Deskripsi** |
| 1 | Register | Fungsi untuk melakukan pendaftaran. |
| 2 | Activate | Fungsi untuk melakukan aktivasi *user via email*. |
| 3 | Login | Fungsi untuk melakukan otentikasi *user*. |
| 4 | Block | Fungsi untuk melakukan *block* terhadap *user*. |
| 5 | Unblock | Fungsi untuk melakukan *unblock* terhadap *user*. |
| 6 | Change Password | Fungsi untuk mengganti *password*. |
| 7 | Delete User | Fungsi untuk menghapus *user*. |

*User* API ini dibangun di atas *framework* Lumen dengan bahasa pemrograman PHP dan memiliki 1 tabel basis data yaitu tabel *Users*. Berikut adalah *Physical Data Model* dari Tabel *Users*:



Gambar 5.7. *Physical Data Model* Tabel *Users*

1. Struktur Data Search Engine

*Search Engine* dibangun di atas *framework* ExpressJS dengan bahasa pemrograman Javascript. *Search Engine* ini menggunakan 2 database yaitu Minio sebagai media penyimpanan *file* dan *Elastic Search* (NoSQL) sebagai penyimpanan metadata dari *file* tersebut.

*Search Engine* ini memiliki beberapa fitur, diantaranya:

Tabel 5.3. Fitur *Search Engine*

| **No.** | **Fitur** | **Deskripsi** |
| --- | --- | --- |
| 1 | Create Bucket | Fungsi untuk membuat *bucket* |
| 2 | Get List Bucket | Fungsi untuk mendapatkan *bucket list* |
| 3 | Is Bucket Exists | Fungsi untuk melakukan pengecekan eksistensi *bucket* |
| 4 | Delete Bucket | Fungsi untuk menghapus *bucket* |
| 5 | Get Objects | Fungsi untuk mendapatkan *list* *file* yang ada di dalam *bucket* tertentu |
| 6 | Get Object Detail | Fungsi untuk mendapatkan *file* secara spesifik |
| 7 | Get Object Info | Fungsi untuk mendapatkan informasi suatu *file* |
| 8 | Upload | Fungsi untuk *upload file* pdf maksimal 10 MB |
| 9 | Search | Fungsi untuk melakukan pencarian terhadap *file* |
| 10 | Move to Bin | Fungsi untuk memindahkan *file* ke dalam *folder Bin* |
| 11 | Restore from Bin | Fungsi untuk mengembalikan *file* dari *Bin* ke lokasi semula |
| 12 | Remove File | Fungsi untuk menghapus *file* secara permanen |
| 13 | Download | Fungsi untuk mengunduh *file* |

Setiap pengguna akan diberikan satu *bucket* khusus untuk menyimpan filenya sehingga setiap pengguna tidak perlu khawatir tentang keamanan *file* yang diunggah. Untuk mencegah terhapusnya *file* secara tidak sengaja oleh pengguna, sistem secara *default* akan menyimpan *file* yang dihapus oleh pengguna ke dalam sebuah *folder* bernama *Bin*. Bila *file* yang ada di dalam *folder Bin* dihapus, barulah sistem akan menghapus *file* tersebut secara permanen.

Metadata *file* disimpan menggunakan *format* JSON ke dalam *Elastic Search*. Berikut adalah struktur data yang digunakan untuk menyimpan metadata *file*:

|  |
| --- |
| {     index: String,     type: String,     id: String,     body: {         name: String,         path: String,         text: String,         size: Number,         lastModified: DateTime,         metaData: {               owner: String         }     }  } |

Gambar 5.8. Struktur data *Elastic search*

Adapun penjelasan dari masing-masing atribut yaitu:

Tabel 5.4. Penjelasan atribut struktur data *Elastic search*

| **No.** | **Atribut** | **Deskripsi** |
| --- | --- | --- |
| 1 | Index | Indeks dari *file* yang diunggah |
| 2 | Type | Tipe *file* yang diunggah |
| 3 | ID | ID file |
| 4 | Body.Name | Nama *file* |
| 5 | Body.Path | Lokasi *file* disimpan |
| 6 | Body.Text | Teks yang berhasil diekstrak dari *file* dengan OCR |
| 7 | Body.Size | Ukuran *file* dalam *Byte* |
| 8 | Body.LastModified | Tanggal terakhir *file* diubah |
| 9 | Body.MetaData.Owner | Berisi *email* dari pemilik *file* (pengguna) |

1. Implementasi

Berikut ini adalah implementasi antarmuka pada sistem yang telah dibangun.

1. Antarmuka Login

Halaman *login* ini adalah halaman yang digunakan pengguna untuk melakukan proses *login* sebelum bisa menggunakan sistem. Pada halaman ini juga terdapat *link* menuju halaman pendaftaran bagi pengguna yang belum terdaftar. Berikut adalah tampilan dari halaman *login*:



Gambar 5.9. Halaman *Login*

1. Antarmuka Register

Halaman *register* ini adalah halaman yang digunakan bagi pengguna yang ingin mendaftar. Berikut tampilan dari halaman *register*:



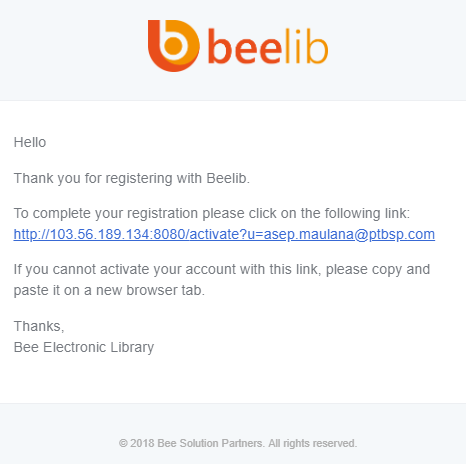
Gambar 5.10. Halaman *Register*

Untuk dapat menggunakan sistem, pengguna harus memiliki *email* aktif, karena pengguna diharuskan melakukan aktivasi akun terlebih dahulu melalui email yang digunakan untuk mendaftar. Bila *link* untuk aktivasi tidak ada di *email*, pengguna bisa mengirimkan *link* kembali melalui halaman *resend*. Berikut adalah tampilan dari halaman *resend*:



Gambar 5.11 Halaman *Resend*

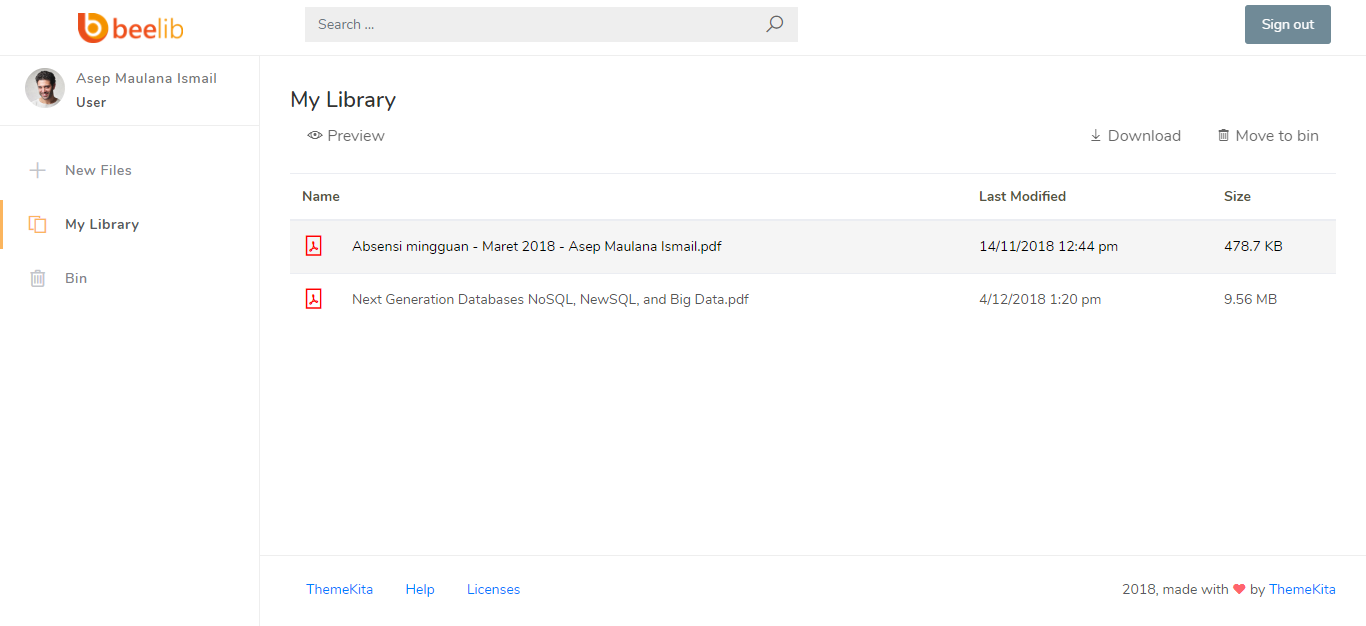
Berikut adalah isi *email* yang dikirim oleh sistem untuk melakukan aktivasi:



Gambar 5.12. *Email Activation*

1. Antarmuka My Libray (Dasbor)

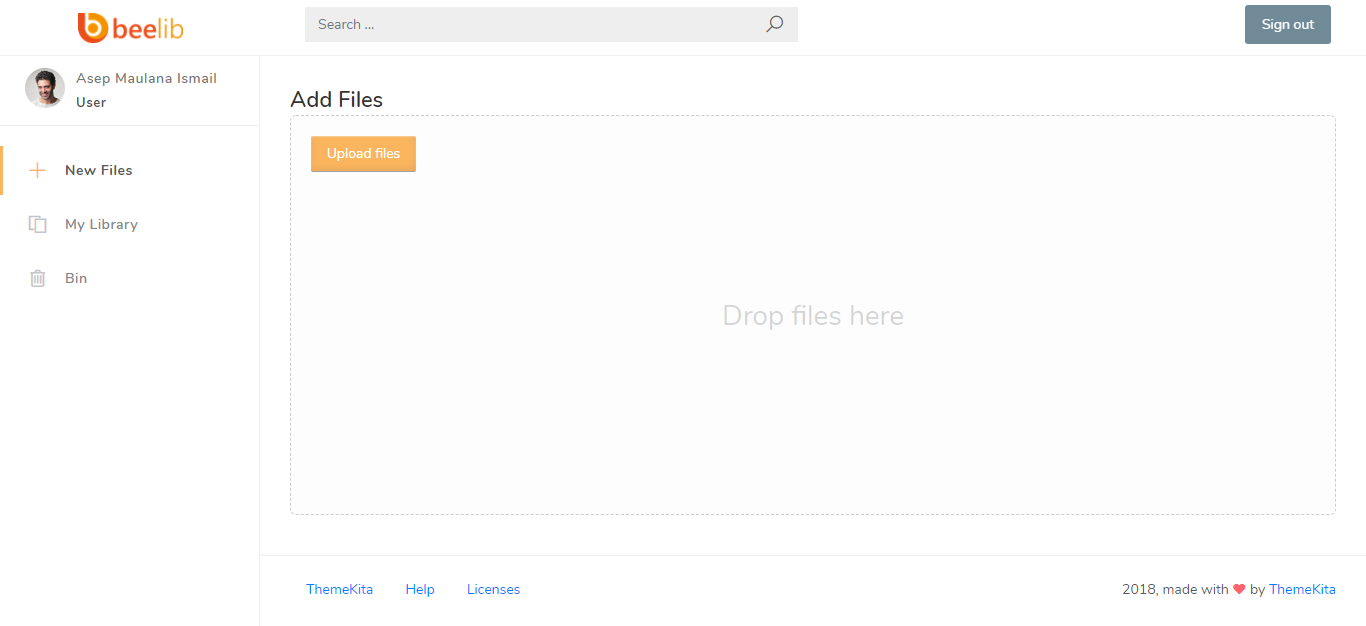
Bila sudah melakukan aktivasi, maka pengguna dapat masuk ke dalam sistem ini. Pengguna akan diarahkan ke halaman *my library* yang berisi daftar *file* yang telah diunggah. Pada halaman ini pengguna dapat melakukan *preview* atau *download* terhadap dokumen. Berikut adalah tampilan dari halaman *my library*:



Gambar 5.13. Halaman *My Library*

1. Antaramuka New Files

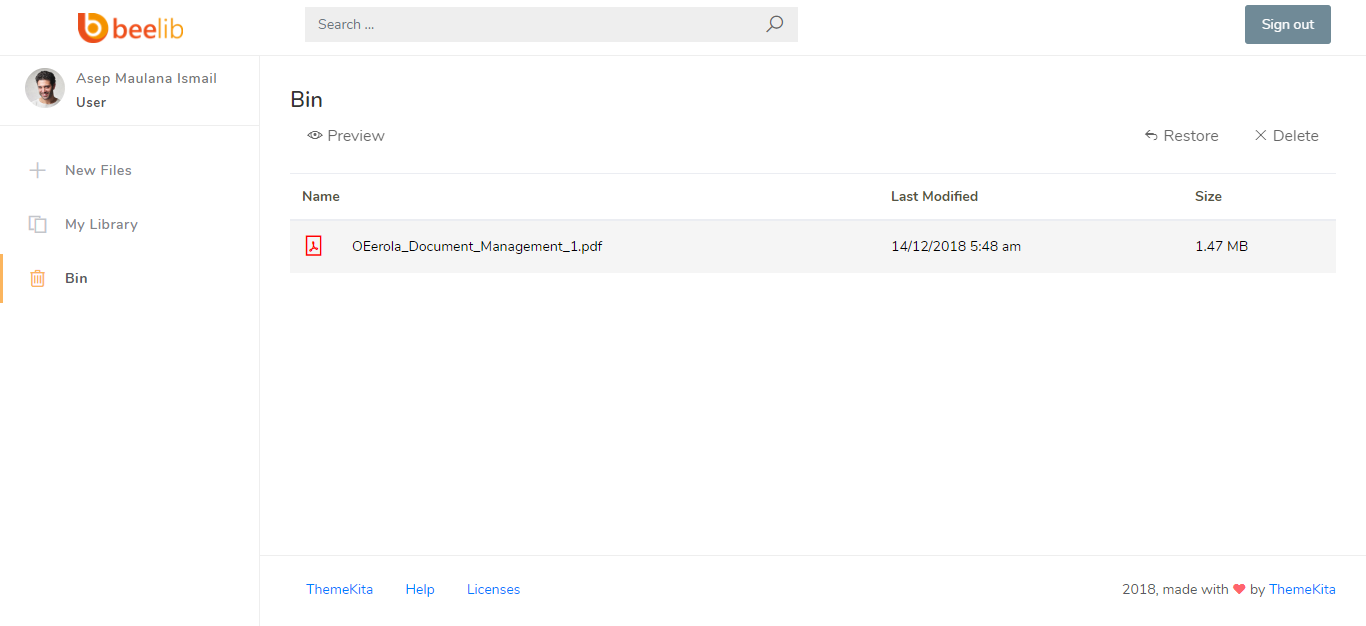
Halaman *new files* adalah halaman yang digunakan pengguna untuk mengunggah dokumen. Berikut adalah tampilan dari halaman *new files*:



Gambar 5.14. Halaman *New Files*

1. Antarmuka Bin

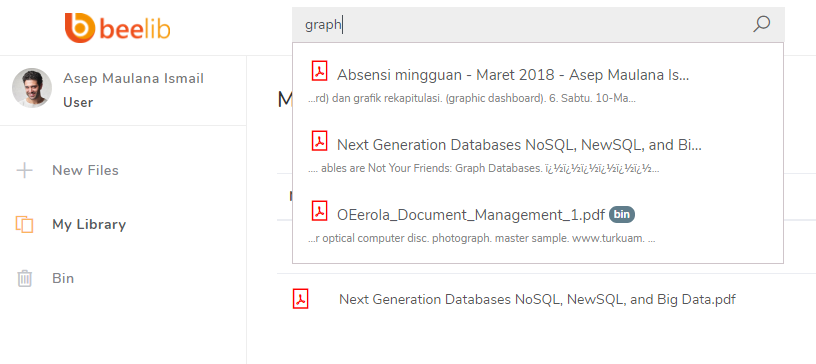
Halaman *bin* adalah halaman dimana dokumen-dokumen yang telah dihapus dari *my library*. Pada halaman ini pengguna dapat mengembalikan *file* ke tempat semula dan juga dapat melakukan penghapusan secara permanen. Berikut adalah tampilan dari halaman *bin*:



Gambar 5.15. Halaman *Bin*

1. Antarmuka Pencarian

Fitur pencarian dapat digunakan oleh pengguna di halaman mana saja, karena field untuk melakukan pencarian disimpan pada *navigation bar*. Pencarian ini juga melibatkan *file* yang berada di *folder bin*. Berikut adalah tampilan ketika pengguna melakukan pencarian terhadap dokumen berdasarkan kata kunci tertentu:



Gambar 5.16. Fitur Pencarian

1. Pengujian

Berikut adalah hasil dari pengujian menggunakan metode *black-box testing*.

5.1.4.1. Pengujian Fitur Register

Pada pengujian ini, fitur *register* pada sistem akan diuji dengan mencoba melakukan registrasi dengan berbagai kemungkinan yang mungkin saja terjadi ketika pengguna melakukan pendaftaran. Adapun tabel hasil pengujian fitur *register* ini adalah sebagai berikut:

Tabel 5.5. Pengujian *Register*

| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| --- | --- | --- | --- |
| Pengguna tidak mengisi *field* Nama | Muncul pesan *error* “*Name can’t be empty*” | Muncul pesan *error* “*Name can’t be empty*” | Berhasil |
| Pengguna tidak mengisi *field Email* | Muncul pesan *error* “*Email can’t be empty*” | Muncul pesan *error* “*Email can’t be empty*” | Berhasil |
| Pengguna tidak mengisi *field Password* | Muncul pesan *error* “*Password can’t be empty*” | Muncul pesan *error* “*Password can’t be empty*” | Berhasil |
| Pengguna tidak mengisi *field Confirm Password* | Muncul pesan *error* “*Confirm Password can’t be empty*” | Muncul pesan *error* “*Confirm Password can’t be empty*” | Berhasil |
| Pengguna mengisi *Email* dengan *format* yang salah | Muncul pesan error “*The email must be a valid email address.*” | Muncul pesan error “*The email must be a valid email address.*” | Berhasil |
| *Password* dan *Confirm password* tidak sesuai | Muncul pesan *error* “*Password & Confirm Password didn't match*” | Muncul pesan *error* “*Password & Confirm Password didn't match*” | Berhasil |
| *Password* kurang dari 8 angka | Muncul pesan *error* “*The password must be at least 8 characters*.” | Muncul pesan *error* “*The password must be at least 8 characters*.” | Berhasil |
| Pengguna menggunakan *email* yang sudah terdaftar | Muncul pesan *error* “*Email is already registered*”. | Muncul pesan *error* “*Email is already registered*.”. | Berhasil |
| Pengguna menggunakan data yang *valid* | - Registrasi berhasil  - Pengguna menerima *email* aktivasi  - Mengarahkan ke halaman *Resend* aktivasi | - Registrasi berhasil  - Pengguna menerima *email* aktivasi  - Mengarahkan ke halaman *Resend* aktivasi | Berhasil |

5.1.4.2. Pengujian Fitur Aktivasi

Pada pengujian ini, fitur aktivasi pada sistem akan diuji. Adapun tabel hasil pengujian fitur aktivasi ini adalah sebagai berikut:

Tabel 5.6. Pengujian Aktivasi

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna menekan tombol *resend activation* | Mengirim ulang *email* aktivasi | Mengirim ulang *email* aktivasi | Berhasil |
| Pengguna menekan *link* aktivasi di *email* | - Pengguna berhasil diaktivasi  - Pengguna dapat melakukan *login* | - Pengguna berhasil diaktivasi  - Pengguna dapat melakukan *login* | Berhasil |

5.1.4.3. Pengujian Fitur Login

Pada pengujian ini, fitur *login* pada sistem akan diuji dengan menggunakan beberapa kombinasi yang mungkin terjadi ketika pengguna melakukan *login*. Adapun tabel hasil pengujian fitur *login* ini adalah sebagai berikut:

Tabel 5.7. Pengujian *Login*

| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| --- | --- | --- | --- |
| Pengguna tidak mengisi *field email* | Muncul pesan *error* “*Email can’t be empty*” | Muncul pesan *error* “*Email can’t be empty*” | Berhasil |
| Pengguna tidak mengisi *field password* | Muncul pesan *error* “*Password can’t be empty*” | Muncul pesan *error* “*Password can’t be empty*” | Berhasil |
| Pengguna mengisi dengan data yang tidak *valid* | Muncul pesan *error* “*Email/Password didn't match*.” | Muncul pesan *error* “*Email/Password didn't match*.”” | Gagal |
| Pengguna mengisi dengan data yang *valid* | Mengarahkan ke halaman *My Library* | Mengarahkan ke halaman *My Library* | Berhasil |

5.1.4.4. Pengujian Fitur Upload

Pada pengujian ini, fitur upload pada sistem akan diuji. Adapun tabel hasil pengujian fitur *upload* ini adalah sebagai berikut:

Tabel 5.8. Pengujian *Upload*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna menggunakan *file* selain PDF | Muncul pesan *error* “*File has an invalid extension. Valid extension(s)*: *pdf*.” | Muncul pesan *error* “*File has an invalid extension. Valid extension(s):* *pdf.”* | Berhasil |
| Pengguna menggunakan *file* dengan ukuran lebih dari 10 MB | Muncul pesan *error* “*File is too large, maximum file size is 10MB*.” | Muncul pesan *error* “*File is too large, maximum file size is 10MB*.”” | Berhasil |
| Pengguna menggunakan *file* PDF dibawah 10 MB | File berhasil diunggah | File berhasil diunggah | Berhasil |

5.1.4.5. Pengujian Fitur Search

Pada pengujian ini, fitur *search* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *search* ini adalah sebagai berikut:

Tabel 5.9. Pengujian *Search*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna menggunakan kata kunci “*database*” (yang ada pada *file* yang sudah diunggah) | Sistem memunculkan *list* rekomendasi dokumen yang sesuai dengan kata kunci | Sistem memunculkan *list* rekomendasi dokumen yang sesuai dengan kata kunci | Berhasil |
| Pengguna menggunakan kata kunci “asdaasdasd” (yang tidak ada pada *file* yang sudah diunggah) | Sistem memunculkan pesan “*No Data Found*” | Sistem memunculkan pesan “*No Data Found*” | Berhasil |
| Pengunna menggunakan kata kunci yang tidak ada pada file name, tetapi ada pada konten | Sistem menampilkan *list* rekomendasi dokumen yang memiliki konten yang sesuai dengan kata kunci. | Sistem menampilkan *list* rekomendasi dokumen yang memiliki konten yang sesuai dengan kata kunci. | Berhasil |

5.1.4.6. Pengujian Fitur Preview

Pada pengujian ini, fitur *preview* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *preview* ini adalah sebagai berikut:

Tabel 5.10. Pengujian *Preview*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna memilih salah satu *file* lalu menekan tombol *preview* | Sistem membuka pratayang *file* yang dipilih pada tab baru *browser* | Sistem membuka pratayang *file* yang dipilih pada tab baru *browser* | Berhasil |

5.1.4.7. Pengujian Fitur Download

Pada pengujian ini, fitur *download* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *download* ini adalah sebagai berikut:

Tabel 5.11. Pengujian *Download*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna memilih salah satu *file* lalu menekan tombol *download* | *File* yang dipilih akan ter-*download* | *File* yang dipilih akan ter-*download* | Berhasil |

5.1.4.8. Pengujian Fitur Move To Bin

Pada pengujian ini, fitur *move to bin* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *move to bin* ini adalah sebagai berikut:

Tabel 5.12. Pengujian *Move To Bin*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna memilih salah satu *file* lalu menekan tombol *move to bin* | *File* yang dipilih akan dipindahkan ke *folder bin* | *File* yang dipilih akan dipindahkan ke *folder bin* | Berhasil |

5.1.4.9. Pengujian Fitur Restore

Pada pengujian ini, fitur *restore* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *restore* ini adalah sebagai berikut:

Tabel 5.13. Pengujian *Restore*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna memilih salah satu file pada *folder bin* lalu menekan tombol *restore* | *File* yang dipilih akan dikembalikan ke *folder My Library* | *File* yang dipilih akan dikembalikan ke *folder My Library* | Berhasil |

5.1.4.10. Pengujian Fitur Delete

Pada pengujian ini, fitur *delete* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *delete* ini adalah sebagai berikut:

Tabel 5.14. Pengujian *Delete*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna memilih salah satu *file* pada *folder bin* lalu menekan tombol *delete* | *File* yang dipilih akan dihapus secara permanen | *File* yang dipilih akan dihapus secara permanen | Berhasil |

5.1.4.11. Pengujian Fitur Logout

Pada pengujian ini, fitur *logout* pada sistem akan diuji. Adapun tabel hasil pengujian fitur *logout* ini adalah sebagai berikut:

Tabel 5.16. Pengujian *Logout*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna menekan tombol *sign out* | Pengguna akan diarahkan ke halaman *login* dan *session* pengguna akan dihapus | Pengguna akan diarahkan ke halaman *login* dan *session* pengguna akan dihapus | Berhasil |

5.1.4.12. Pengujian Keamanan

Pada pengujian ini, sistem akan diuji untuk mengetes apakah sebuah akun pengguna dapat melihat file pengguna lain. Adapun tabel hasil pengujian adalah sebagai berikut:

Tabel 5.17. Pengujian Keamanan

| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| --- | --- | --- | --- |
| Dua pengguna mengunggah 2 file dengan nama yang sama | File tidak konflik, unggah berhasil | File tidak konflik, unggah berhasil | Berhasil |
| Pengguna A masuk kedalam sistem dan mengunggah file, lalu keluar, dan pengguna B masuk kedalam sistem | Pengguna B tidak bisa melihat file Pengguna A | Penguna B tidak bisa melihat file Pengguna A | Berhasil |
| Pengguna tanpa melakukan *login* langsung mengakses halaman *My* *Library* | Sistem melakukan *redirect* ke halaman *login* | Sistem melakukan *redirect* ke halaman *login* | Berhasil |

1. Fitur Gagal

Terdapat satu fitur yang gagal dari fitur *login* yaitu ketika pengguna memasukan *data* yang tidak *valid* ke dalam *form login*, sistem tidak dapat memunculkan pesan *error*.

Karena memiliki kompleksitas yang kecil, maka kesalahan ini bisa langsung diperbaiki. Adapun hasil pengujian setelah dilakukan perbaikan terhadap fitur *login* adalah sebagai berikut:

Tabel 5.18. Pengujian Kembali Fitur *Login*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test Case*** | ***Expected Result*** | ***Actual Result*** | **Kesimpulan** |
| Pengguna mengisi dengan *data* yang tidak *valid* | Muncul pesan *error* “*Email/Password didn't match*.” | Muncul pesan *error* “*Email/Password didn't match*.”” | Berhasil |

1. Kesimpulan Pengujian

Pada proses pengujian pertama, terdapat satu bug yaitu aplikasi tidak memunculkan pesan *error* ketika pengguna memasukan data yang tidak *valid* ke dalam *form login*. Setelah dilakukan perbaikan dan dilakukan pengujian ulang terhadap semua fitur, dapat disimpulkan bahwa fitur aplikasi pada *sprint* 1 ini 100% telah berjalan baik.

BAB VI  
PENUTUP

1. Kesimpulan

Dari semua proses pembuatan aplikasi *Document Management System* (DMS) ini dapat diambil kesimpulan:

1. Sistem mampu menampilkan dokumen yang relevan dengan kata kunci pencarian. Hal ini dapat dibuktikan dengan melihat hasil pengujian pada Tabel 5.9 tentang pengujian fitur *Search*.
2. Sistem mampu membaca konten dari *file* dokumen. Hal ini dapat dibuktikan dengan melihat hasil pengujian pada Tabel 5.9 tentang pengujian fitur *Search*. Dari pengujian tersebut didapatkan bahwa pencarian dapat dilakukan tidak hanya kepada nama file, namun kepada isi dari file tersebut.
3. Sistem mampu mengamankan *file* pengguna dari akses pengguna lain. Hal ini dapat dibuktikan dengan melihat hasil pengujian pada Tabel 5.17. Dari pengujian tersebut didapatkan bahwa setiap file hanya bisa diakses oleh pemilik file tersebut. Dan meskipun pengguna yang berbeda mengunggah file dengan nama file yang sama, hal tersebut tidak akan membuat file tercampur.
4. Saran

Adapun saran yang diajukan untuk pengembangan aplikasi *Document Management System* (DMS) ini yaitu:

1. Pengembangan dapat dilanjutkan ke tahap selanjutnya yaitu *Sprint* 2 untuk menambahkan fungsi *Optical Character Recognition* (OCR) agar dokumen yang didapatkan dari hasil pemindaian (*scan*) dapat terindeks.
2. Pada pengembangan selanjutnya disarankan untuk menambah jenis *file* baru seperti docx, xlsx, jpg, png, gif, dan bmp. Selain itu juga disarankan untuk menambah kapasitas maksimal *file* yang bisa diunggah ke dalam aplikasi DMS.

DAFTAR PUSTAKA

1. O`Brien, J.A., “*Introduction to Information Systems: Essentials for the e-business enterprise*”, McGraw-Hill, 2003
2. Layton, Mark C., “*Scrum for Dummies*”, John Wiley & Sons, Inc, 2015
3. \_ \_ \_ \_ \_ \_, “*Company Profile*”, PT. Bee Solution Partners, 2018
4. Turban, Efraim, Linda Volonino, Gregory Wood, “*Information Technology for Management Advancing Sustainable, Profitable Business Growth 9th Edition*”, John Wiley & Sons, Inc, 2013
5. Rouse, Margaret, “*What is file storage*?”, TechTarget, Mei 2016, < https://searchstorage.techtarget.com/definition/file-storage> [diakses 22 Oktober 2018]
6. Rouse, Margaret, “*What is Scrum*?”, TechTarget, Februari 2017, < https://searchsoftwarequality.techtarget.com/definition/Scrum> [diakses 22 Oktober 2018]
7. Chen, Lianping, “*Microservices: Architecting for Continuous Delivery and DevOps*”, The IEEE International Conference on Software Architecture (ICSA 2018), 2018
8. Software, Redstone, “Black-box vs. White-box Testing: Choosing the Right Approach to Deliver Quality Applications”, Redstone Software, 2008