KURIR DATA DIGITAL: SISTEM BERBAGI INFORMASI DIGITAL PEDESAAN

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

I KETUT ADI SUHENDRA

NIM. 0910681008

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

PROGRAM STUDI TEKNIK INFORMATIKA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2013

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LEMBAR PERSETUJUAN

KURIR DATA DIGITAL: SISTEM BERBAGI INFORMASI DIGITAL PEDESAAN

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

I KETUT ADI SUHENDRA

NIM. 0910681008

Skripsi ini telah diperiksa dan disetujui oleh dosen pembimbing pada tanggal

9 Desember 2013

Dosen Pembimbing I	Dosen Pembimbing II
Achmad Basuki, ST., MMG., Ph.D.	Eko Sakti Pramukantoro, S.Kom., M.Kom.
NIP. 19741118 200312 1 002	NIK. 860805 06 1 1 0252

LEMBAR PENGESAHAN

KURIR DATA DIGITAL: SISTEM BERBAGI INFORMASI DIGITAL PEDESAAN

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

I KETUT ADI SUHENDRA

NIM. 0910681008

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal

20 Desember 2013

Penguji I	Penguji II
Kasyful Amron, ST., M.Sc.	Ir. Heru Nurwasito, M.Kom.
NIP. 19750803 200312 1003	NIP. 19650402 199002 1001

Penguji III

Sabriansyah Rizqika Akbar, ST., M.Eng.
NIK. 820809 0611 0084

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.Si.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : I Ketut Adi Suhendra
NIM : 0910681008
Program Studi : Teknik Informatika
Judul Skripsi : Kurir Data Digital: Sistem Berbagi Informasi Digital Pedesaan

Saya yang bertanda tangan dibawah ini menyatakan bahwa :

Isi dari Skripsi ini adalah benar-benar karya sendiri dan tidak terdapat karya yang pernah di ajukan oleh orang lain atau kelompok lain untuk memperoleh gelar akademis di suatu Institusi Pendidikan, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Demikian pernyataan ini saya buat dengan sesungguhnya. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti terdapat penyimpangan, maka saya akan bersedia menerima sanksi sesuai peraturan yang berlaku di perguruan tinggi ini.

Malang, November 2013

I Ketut Adi Suhendra
NIM. 0910681008

KATA PENGANTAR

Dengan nama Ida Sang Hyang Widi Wasa. Segala puji syukur saya panjatkan karena atas rahmatNya-lah skripsi dengan judul "Kurir Data: Sistem Berbagi Informasi Digital Pedesaan" ini dapat terselesaikan dengan baik.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan baik lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Bapak Ir. Sutrisno, M.T., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Himawat Aryadita, S.T., M.Sc., dan Bapak Eddy Santoso, S.Kom. selaku Ketua, Wakil Ketua 1, Wakil Ketua 2, dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
2. Bapak Drs. Marji, M.T. dan Bapak Issa Arwani, S.Kom., M.Sc. selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
3. Bapak Achmad Basuki, ST., MMG., Ph.D. dan Bapak Eko Sakti Pramukantoro S.Kom., M.Kom. selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis, sehingga dapat menyelesaikan skripsi ini.
4. Bapak Arief Andy Soebroto, ST., M.Kom. selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
5. Bapak Eko Sakti Pramukantoro S.Kom., M.Kom. selaku Ketua Laboratorium Jaringan Komputer.
6. Seluruh Dosen dan Staf Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas kesediaan membagi ilmu dan bantuan yang telah diberikan.
7. Kedua orang tua beserta kakak-kakak dan keluarga besar yang selalu memberikan bantuan, dukungan, dan doa sebagai pemberi semangat untuk terus maju menyelesaikan skripsi.

8. Teman-temanku indra, yogik, arfian, dewa, yana, dan semua teman-teman PTIIK angkatan 2009 serta teman-temanku lainnya terima kasih atas kesempatan yang diberikan untuk berbagi segala pengetahuan dan cerita baik suka maupun duka, serta bantuan dan dorongan semangat selama perkuliahan dan proses mengerjakan skripsi
9. Ni Made Shrivijayanthi yang telah memberikan saya motivasi, semangat, dan dukungan selama ini.
10. Seluruh pihak yang telah membantu dalam penyelesaian penulisan skripsi ini yang tidak dapat penulis sebut satu persatu.

Penulis yakin skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat diharapkan. Semoga skripsi ini membawa manfaat bagi kita semua.

Malang, November 2013

Penulis

ABSTRAK

Information sharing merupakan hal yang menjadi kebutuhan dasar bagi masyarakat di era globalisasi ini. Akses terhadap Internet menjadi sangat penting untuk menjembatani penyebaran informasi. Ketersediaan akses Internet di pedesaan dan perkotaan di Indonesia tidaklah merata, terutama di daerah pedesaan dikarenakan penyediaan infrastruktur yang cukup mahal atau letak geografis yang sulit dijangkau. Penyebaran Internet yang tidak merata menyebabkan adanya kesenjangan informasi dan komunikasi (*digital divide*) antara pedesaan dan perkotaan di Indonesia.

Penelitian ini menawarkan sebuah sistem berbagi informasi digital tanpa adanya koneksi Internet. Sistem ini dibangun berdasarkan konsep *Delay-Tolerant Network* (DTN). Namun, ketiadaan koneksi Internet permanen di pedesaan menyebabkan mekanisme *store and forward* pada DTN tidak dapat berfungsi, sehingga dalam sistem ini mengusulkan mekanisme kurir data. Dengan menggunakan jasa kurir data, maka kendala geografis tidak lagi menjadi penghambat penyebaran informasi. Hasil evaluasi sistem dan skenario penggunaan sistem dari penelitian ini dapat mengatasi *digital divide* antara pedesaan dan perkotaan dan menjadi solusi alternatif terhadap teknologi yang sudah ada saat ini.

Kata kunci*: *DTN, Internet, pedesaan

repository.ub.ac.id

ABSTRACT

Information sharing is a primary needs for society in this globalization. Internet access has become very important to facilitate the information sharing. Internet access is not always available in rural area in Indonesia due to the expensive deployment of Internet infrastructure or inaccessible geographical reason.The unequal availability of Internet access between rural and urban area is causing the digital divide.

This research proposes a digital information sharing system without Internet connection. The proposed system is built based on the concept of Delay-Tolerant Network (DTN). However, the inexistence of Internet connectivity in rural area is causing the store and forward mechanism does not working properly, hence, this system proposed a courier data mechanism. By using data courier service, inaccessible geographical area issue is not the barrier anymore. The proposed system was evaluated in several usage scenarios and it is proved that it works perfectly. Therefore, the proposed system can overcome the digital divide problem between rural and urban area and it becomes an alternative solution to existing technology today.

Keywords: DTN, internet, rural

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

KATA PENGANTAR i
ABSTRAK iii
ABSTRACT iv
DAFTAR ISI v
DAFTAR GAMBAR vii
DAFTAR TABEL ix
DAFTAR LAMPIRAN x
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan 3
1.5 Manfaat 4
1.6 Sistematika Penulisan 4
BAB II KAJIAN PUSTAKA DAN DASAR TEORI 5
2.1 Data Digital 5
2.2 Pengiriman Data 6
2.2.1 Pengiriman Data pada Internet 7
2.2.2 Pengiriman Data pada DTN 8
2.2.3 Pengiriman Data dengan Kurir Data 9
2.3 Pengolahan Data 10
BAB III METODE PENELITIAN 11
3.1 Studi Literatur 12
3.2 Metode Pengambilan Data 12
3.3 Analisis Kebutuhan 13
3.3.1 Kebutuhan Aplikasi 13
3.3.2 Kebutuhan Lingkungan Sistem 13
3.4 Perancangan Sistem 14
3.5 Implementasi 15
3.6 Pengujian dan Analisis 17
3.7 Kesimpulan dan Saran 18
BAB IV PERANCANGAN 19

4.1 Perancangan 19
4.1.1 Desain Arsitektur Sistem 19
4.1.2 Fitur aplikasi 22
4.1.3 Desain Aplikasi 24
4.2 Implementasi 36
4.2.1 Lingkungan Implementasi 37
4.2.2 Batasan Implementasi 37
4.2.3 Implementasi Metode 37
4.2.4 Implementasi Antarmuka 45
BAB V PENGUJIAN DAN ANALISIS 48
5.1 Pengujian 48
5.1.1 Pengujian Kebutuhan Fungsional 48
5.1.2 Pengujian Kebutuhan Non-Fungsional 56
5.2 Analisis 57
5.2.1 Analisis Kebutuhan Fungsional 57
5.2.2 Analisis Kebutuhan Non-Fungsional 58
BAB VI PENUTUP 60
6.1 Kesimpulan 60
6.2 Saran 61
DAFTAR PUSTAKA DP-1
LAMPIRAN 1 ..... L-1
LAMPIRAN 2 ..... L-2
LAMPIRAN 3 ..... L-3

DAFTAR GAMBAR

Gambar 2.1. Model Komunikasi *Client* dan *Web Server* 7
Gambar 2.2. Mekanisme *Store and Forward* pada VDTN 8
Gambar 3.1. Alur Metode Penelitian 11
Gambar 3.2. Alur Implementasi Lingkungan Sistem 15
Gambar 3.3. Alur Implementasi Aplikasi Sistem 16
Gambar 3.4. Alur Pengujian Fungsional Aplikasi *User* dan Kurir Data 17
Gambar 3.5. Alur Pengujian dan Analisis Sistem 18
Gambar 4.1. Skenario Sistem 19
Gambar 4.2. Sistem Kerja Kurir Data 20
Gambar 4.3. Alur Skenario Sistem 20
Gambar 4.4. Arsitektur LAN pada Sink POA 21
Gambar 4.5. Use Case: *User* 24
Gambar 4.6. Use Case: Kurir Data di Sink POA 25
Gambar 4.7. Use Case: Kurir Data di Source POA 26
Gambar 4.8. Pola MVC pada File Browser 27
Gambar 4.9. Pola MVC pada File Collector 27
Gambar 4.10. ER Diagram Basis Data File Browser 28
Gambar 4.11. Sequence Diagram: *User* 29
Gambar 4.12. Sequence Diagram: Kurir Data di Sink POA 30
Gambar 4.13. Sequence Diagram: Kurir Data di Source POA 32
Gambar 4.14. Activity Diagram: *User* 34
Gambar 4.15. Activity Diagram: Kurir Data di Sink POA 35
Gambar 4.16. Activity Diagram: Kurir Data di Source POA 36
Gambar 4.17. Prosedur insert_files() 38
Gambar 4.18. Prosedur delete_files() 39
Gambar 4.19. Prosedur insert_request() 40
Gambar 4.20. Prosedur delete_request() 41
Gambar 4.21. Prosedur search() 42
Gambar 4.22. Prosedur google_search() 43

Gambar 4.23. Prosedur copy_file() 44
Gambar 4.24. Prosedur download_file() 45
Gambar 4.25. Antarmuka Utama File Browser 46
Gambar 4.26. Antarmuka Utama File Collector 46
Gambar 5.1. Antarmuka: Hasil Pencarian File 48
Gambar 5.2. Antarmuka: Baca File 49
Gambar 5.3. Antarmuka: Tambah Permintaan Berhasil 50
Gambar 5.4. Antarmuka: Permintaan Sudah Ada 50
Gambar 5.5. Antarmuka: Permintaan Tidak Sesuai Kosa Kata 51
Gambar 5.6. Antarmuka: Pemindahan File 52
Gambar 5.7. Antarmuka: Penerimaan Permintaan 52
Gambar 5.8. Antarmuka: File Dibuka Aplikasi Lain 53
Gambar 5.9. Antarmuka: Penyalinan File 54
Gambar 5.10. Antarmuka: Free Space Memory Tidak Cukup 55

DAFTAR TABEL

Tabel 5.1. Hasil Pengujian Validasi File Browser.....51
Tabel 5.2. Hasil Pengujian Validasi File Collector.....56

DAFTAR LAMPIRAN

Lampiran 1 Statistik Pengguna Internet di Indonesia……..... L-1
Lampiran 2 Data Akses Internet di Kelurahan di Indonesia……... L-2
Lampiran 3 Detail Desain Aplikasi……...……...……L-3

BAB I
PENDAHULUAN

1.1 Latar Belakang

Information sharing merupakan hal yang menjadi kebutuhan dasar bagi masyarakat di era globalisasi khususnya informasi digital. Akses terhadap Internet menjadi sangat penting untuk menjembatani penyebaran sebuah informasi, tetapi tidak semua kalangan masyarakat di Indonesia dapat menikmatinya. Dari hasil survey oleh MarkPlus Insight Netizen [DWS-13], pada tahun 2012 jumlah pengguna Internet di Indonesia mencapai 61 juta orang, meningkat 6 juta dari tahun 2011 sebesar 55 juta orang. Dengan demikian, total pengguna Internet di Indonesia hanya 23,5% dari total populasi.

Akses Internet belum tersedia di semua daerah terutama di daerah pedesaan dikarenakan penyediaan infrastruktur yang cukup mahal atau letak geografis yang sulit dijangkau. Data dari Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) [APJ-12], pada tahun 2012 menunjukkan dari 69.957 total kelurahan yang ada di seluruh Indonesia, kelurahan dengan akses kabel Telkom di pedesaan sebanyak 23,03% dan kota sebanyak 89,32%. Data kelurahan yang memiliki warnet di pedesaan adalah sebanyak 0,62% dan kota sebanyak 14,39%. Persentase tersebut berdasarkan 57.667 total kelurahan di pedesaan dan 12.290 total kelurahan di perkotaan. Berdasarkan data, dapat dilihat perbandingan yang sangat mencolok pada ketersedian akses Internet antara daerah pedesaan dan perkotaan. Hal ini menyebabkan adanya kesenjangan informasi dan komunikasi (*digital divide*) antara pedesaan dan perkotaan di Indonesia, padahal bila kita melihat fakta, misalkan pada beberapa kasus, masyarakat kota lebih mengetahui teknologi pertanian dan peternakan saat ini, padahal yang menerapkan adalah masyarakat pedesaan.

Saat ini memang sudah ada beberapa cara dilakukan untuk menyediakan akses terhadap informasi digital di wilayah pedesaan di Indonesia, misalnya Proyek Palapa Ring [PRD-13] berupa pembangunan serat

optik nasional di seluruh Indonesia sepanjang 36.000 kilometer yang akan menjangkau 440 kota/kabupaten di seluruh Indonesia, tetapi masih dalam tahap pembangunan. Selain itu, untuk saat ini sudah diterapkan juga program Internet Masuk Desa [IMD-13] oleh Depkominfo. Program Internet Masuk Desa cocok diterapkan di daerah pedesaan yang dapat dijangkau oleh alat transportasi dan tidak memiliki infrastruktur jaringan sama sekali. Program Internet Masuk Desa dalam implementasinya masih banyak memiliki kendala diantaranya biaya operasional yang masih tinggi dan masih rendahnya masyarakat pedesaan dalam literasi mengenai perkembangan teknologi digital saat ini.

Beberapa daerah ada yang menerapkan teknologi *Delay-Tolerant Network* (DTN) [FKD-03] yang mengatasi ketiadaan koneksi *end-to-end*. Pada aplikasi berbasis DTN, ada mekanisme *store and forward* pada tiap *relay node* untuk pengiriman data dan penyimpanan data sementara. Pada daerah terpencil seperti pedesaan, *Vehicular Delay-Tolerant Network* (VDTN) [JVD-13] menjadi solusi yang umum. VDTN memungkinkan pengiriman data pada daerah yang memiliki akses transportasi yang cukup baik, tetapi apabila diimplementasikan pada daerah yang terbilang sangat terpencil, maka mekanisme *store and forward* pada VDTN ini tidak akan dapat bekerja secara maksimal. Jadwal alat transportasi yang tidak konsisten dan kondisi geografis yang sulit dijangkau dapat menyebabkan waktu penyimpanan data (*bundle*) pada *relay node* melebihi batas yang didefinisikan oleh *Bundle Protocol Agent* (BPA), sehingga data mengalami *expired* dan dibuang sebelum mencapai *relay node* selanjutnya.

Sistem yang dibangun dalam penelitian ini menerapkan skenario seperti DTN, tetapi menghilangkan mekanisme *store and forward* dan menggantinya dengan kurir data. Kurir data adalah seseorang yang bertugas mencari dan membawa kembali informasi digital yang dibutuhkan. Dengan menggunakan jasa kurir data, maka kondisi geografis tidak terlalu menjadi masalah. Hasil dari penelitian ini diharapkan dapat mengatasi *digital divide* antara pedesaan dan perkotaan dan menjadi solusi lain terhadap teknologi yang sudah ada saat ini. Sistem ini nantinya diharapkan menjadi sistem

berbagi informasi digital yang murah dan mudah digunakan dengan kualitas yang dapat diandalkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka dirumuskan beberapa permasalahan:

- Bagaimana desain dan implementasi prototipe sistem berbagi informasi digital pada daerah terpencil?
- Bagaimana skenario penggunaan dan evaluasi prototipe sistem berbagi informasi digital di daerah terpencil?

1.3 Batasan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka batasan masalahnya adalah sebagai berikut:

- Informasi yang dapat diakses oleh pengguna hanya informasi yang terdapat pada file dengan format PDF.
- Informasi dapat dibaca secara langsung oleh pengguna dengan menggunakan setiap *end device* (tablet, *smartphone*, PC, dll) yang memiliki *browser* yang mendukung PDF Reader.
- Kurir data dianggap memiliki pengetahuan dasar penggunaan perangkat lunak.
- Kurir data memiliki akses ke sumber data dengan alat transportasi secara berkala.

1.4 Tujuan

Membuat sebuah sistem yang berfungsi menyediakan/menjembatani layanan akses informasi digital di daerah terpencil dengan ketiadaan jaringan komunikasi data dan Internet. Sistem ini diterapkan di wilayah pedesaan, sehingga menjadi solusi atas kesenjangan informasi digital antara pedesaan dan perkotaan.

1.5 Manfaat

- Setiap orang di pedesaan tidak perlu lagi pergi ke kota atau sumber data lainnya untuk mendapatkan informasi digital yang diinginkannya, sehingga murah dari segi biaya.
- Dapat meningkatkan Sumber Daya Manusia di pedesaan dari segi ekonomi, sosial, maupun budaya dengan kemudahan akses informasi digital.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I PENDAHULUAN

Memuat latar belakang permasalahan, identifikasi dan pembatasan masalah, rumusan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Menguraikan semua teori dasar dan teori penunjang yang berkaitan dengan pembuatan sistem.

BAB III METODOLOGI PENELITIAN

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, perancangan, implementasi, pengujian, dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV PERANCANGAN

Membahas analisis kebutuhan, arsitektur dari aplikasi, serta implementasi aplikasi.

BAB V PENGUJIAN DAN ANALISIS

Memuat proses dan hasil pengujian terhadap metode yang telah direalisasikan.

BAB VI PENUTUP

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian metode untuk pengembangan lebih lanjut.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Data Digital

Data dalam bentuk informasi digital (file) adalah data yang tersimpan pada komputer atau media penyimpanan digital. File memiliki banyak tipe tergantung dari aplikasi yang memprosesnya dan setiap tipe file memiliki fungsi yang berbeda. File yang umum digunakan dalam penyediaan informasi saat ini adalah PDF [ASI-06]. PDF merupakan format file yang mewakili dokumen secara independen dari aplikasi perangkat lunak, perangkat keras, dan sistem operasi yang membuatnya, sehingga isinya tidak akan berubah apabila filenya dibuka di *platform* lain. PDF mendukung keterangan dari berbagai hal seperti catatan teks, *hypertext link*, *markup*, lampiran file, suara, dan film. Isi yang terkandung dalam file PDF biasanya berupa *paper*, majalah, dokumentasi, dan lain sebagainya.

Selain file PDF, ada pula file XML [JMX-03] yang sering kali diintegrasikan dalam pengolahan informasi digital. XML adalah sebuah format dokumen yang digunakan untuk memberi label, mengkategorikan, dan mengatur informasi dengan cara tertentu. Dokumen XML dapat disusun dengan struktur data sesuai kebutuhan. Beberapa keuntungan file XML adalah sebagai berikut:

1. Mendukung klasifikasi informasi, sehingga data mudah untuk dicari dan diambil.
2. Memberi aturan pada data.
   - Menjamin keakuratan informasi yang dikumpulkan.
   - Menjamin bahwa data yang diterima adalah data yang tepat sesuai keperluan.
3. Menampilkan informasi dengan berbagai macam cara.
4. Menggunakan data yang sama pada berbagai *platform*.

Aplikasi berbasis *web* maupun *desktop* biasanya menggunakan beberapa file konfigurasi. Pada perancangan aplikasi, file XML sangat cocok digunakan sebagai file konfigurasi aplikasi karena strukturnya yang baik.

Dalam penyebaran informasi digital, ada beberapa tipe file yang biasa diakses oleh pengguna. Selain file PDF, ada pula file *text* dan file *Hypertext Markup Language* (HTML) [BKC-05]. File *text* biasanya hanya mengandung catatan, *source code* program, konfigurasi sistem, dan lain sebagainya. File HTML terdiri dari berbagai jenis objek seperti file PDF, tetapi masing-masing objek merupakan tautan dari berbagai sumber. Sebuah file HTML dapat mengandung berbagai informasi, tetapi sering kali tidak jelas orisinalitasnya.

2.2 Pengiriman Data

Jaringan komputer adalah kumpulan komputer yang dapat berkomunikasi satu sama lain untuk berbagi sumber daya. Sumber daya yang dikirimkan pada sebuah jaringan komputer dapat berupa data digital seperti dokumen dan aplikasi.

Proses pengiriman data pada jaringan komputer akan melewati banyak perangkat yang disebut dengan *node*. *Node* yang bertindak sebagai pengirim data disebut dengan *source*, sedangkan *node* yang bertindak sebagai penerima data disebut dengan *destination*. Pengiriman data dari *source* ke *destination* akan melewati *node* perantara seperti *router*, *switch*, dan *bridge* yang dihubungkan oleh media transmisi data berupa kabel atau nirkabel.

Model interaksi yang terjadi dalam pengaksesan sumber daya di jaringan komputer dibagi menjadi dua, yaitu model *client-server* dan *peer-to-peer*. Pada model *client-server*, sebuah *server* didedikasikan sebagai penyedia layanan untuk para *client*, sedangkan pada model *peer-to-peer*, sebuah komputer dapat menjadi *server* maupun *client*.

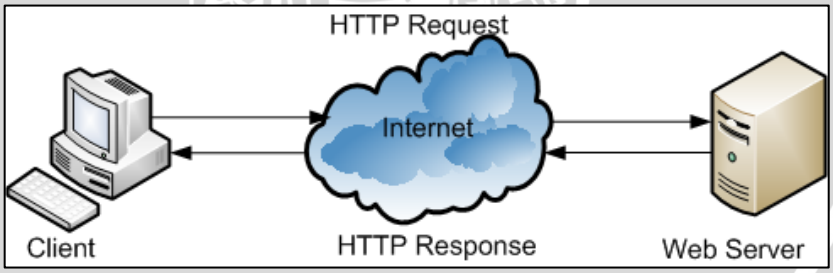
Komunikasi data yang terjadi pada jaringan komputer menggunakan mekanisme pengendalian aliran data untuk menjamin data dari *source* sampai di *destination*. Mekanisme pengendalian aliran data ini berbeda pada setiap teknologi di jaringan komputer. Internet menggunakan sebuah standar komunikasi data yang disebut dengan arsitektur *Transmission Control*

Protocol/Internet Protocol (TCP/IP) [BKC-05]. Selain TCP/IP, ada juga arsitektur *Delay-Tolerant Network* (DTN) [FKD-03].

2.2.1 Pengiriman Data pada Internet

Pada proses pengiriman data digital di Internet, seorang pengguna dapat menggunakan salah satu *application layer* TCP/IP, yaitu *Hypertext Transfer Protocol* (HTTP) [BKC-05] untuk mengakses sumber daya yang ada pada *server* target melalui *Uniform Resource Locator* (URL). HTTP adalah sebuah protokol dengan model *client-server* yang dapat digunakan oleh aplikasi *web* maupun aplikasi *desktop*. Aplikasi *web* yang bertindak sebagai *server* memerlukan sebuah *web server* sebagai *platform* agar dapat diakses oleh *client* melalui *web browser*. *Web server* dapat dipilih sesuai performa aplikasi yang diinginkan. Salah satu jenis *web server* adalah Apache Web Server [AWS-07]. Beberapa kelebihan Apache Web Server antara lain:

1. Merupakan *freeware*.
2. Mudah dalam proses instalasi.
3. Dapat beroperasi di berbagai *platform*.
4. Mudah dikonfigurasi.



Gambar 2.1. Model Komunikasi *Client* dan *Web Server*

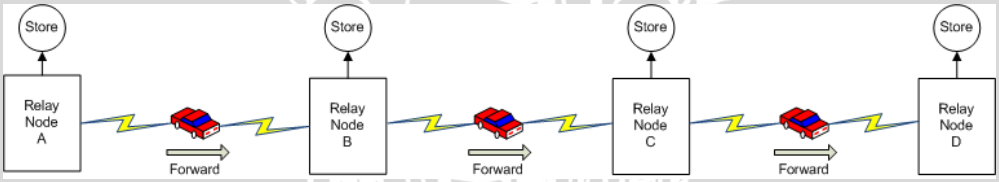
Gambar 2.1 menunjukkan model komunikasi antara *client* dan *web server* yang terjadi di Internet. *Client* melakukan *request* kepada *web server* melalui protokol HTTP pada *port* 80 dan *web server* membalas dengan mengirimkan *response* kepada *client* dengan protokol HTTP pada *port* yang

disediakan aplikasi *client*. Hal ini juga berlaku saat *client* mengakses *web server* yang ada di jaringan lokal.

2.2.2 Pengiriman Data pada DTN

Pada TCP/IP, ada aturan yang tidak memperbolehkan adanya *delay* yang terlalu tinggi pada komunikasi antar *node*. TCP/IP akan dapat bekerja apabila ada koneksi *end-to-end* antar *node* yang selalu tersedia secara dua arah. Selain TCP/IP, ada arsitektur lain yang diterapkan pada jaringan komunikasi data, yaitu *Delay-Tolerant Network* (DTN). DTN memungkinkan terjadinya proses pengiriman data dengan *delay* yang cukup tinggi dan ketiadaan koneksi *end-to-end*.

Perbedaan antara pengiriman data menggunakan TCP/IP dan DTN terletak pada mekanise pengendalian aliran pengiriman data. Data yang dikirim dengan menggunakan TCP/IP, akan melewati banyak *node* selama *node* tersebut tersedia tanpa ada proses penyimpanan data, sedangkan pada DTN, terjadi mekanisme *store and forward*, dimana data terlebih dahulu disimpan di *node* yang dilewatinya sebelum dikirimkan ke *node* selanjutnya.



Gambar 2.2. Mekanisme *Store and Forward* pada VDTN

Daerah pedesaan biasanya menerapkan *Vehicular Delay-Tolerant Network* (VDTN) [JVD-13]. Gambar 2.2 menunjukkan mekanisme *store and forward* pada VDTN. VDTN bekerja dengan memanfaatkan alat transportasi seperti kereta dan mobil sebagai pengantar paket data (*bundle*) ke *node* tujuan. VDTN membutuhkan jadwal perjalanan kendaraan secara konsisten agar *bundle* dapat mencapai *node* penyimpanan data, atau yang disebut dengan *relay node*. *Relay node* ditempatkan di beberapa titik yang

akan dilewati oleh kendaraan. Ketika sebuah kendaraan mendapatkan paket *bundle*, maka *Bundle Protocol Agent* (BPA) akan menentukan *relay node* yang menjadi tujuan pengiriman paket selanjutnya. Apabila ada kendaraan yang melewati jalur lain atau mengalami masalah saat beroperasi, maka paket *bundle* kemungkinan tidak mencapai *relay node* dalam waktu yang telah ditentukan, sehingga *bundle* mengalami *expired* dan paket harus dibuang.

VDTN memungkinkan untuk diterapkan di daerah pedesaan dengan jadwal alat transportasi yang konsisten. Pada daerah yang sangat terpencil, biasanya masih ada permasalahan alat transportasi yang juga dikarenakan permasalahan letak geografis. Hal ini menyebabkan mekanisme *store and forward* pada VDTN tidak dapat bekerja secara maksimal, sehingga VDTN sangat sulit diterapkan pada daerah yang sangat terpencil.

2.2.3 Pengiriman Data dengan Kurir Data

Pada proses pengiriman data dengan adanya koneksi *end-to-end* (seperti pada jaringan lokal dan Internet) maupun pada DTN, informasi yang dipertukarkan merupakan pecahan-pecahan data (*data chunks*) yang dikirimkan oleh sumber (*source*) ke tujuan (*destination/sink*). Data utuh yang ada pada *source* dipecah dan dikirimkan ke *destination*, dimana pecahan data tersebut dikumpulkan kembali menjadi data yang utuh.

Pada sistem berbagi informasi digital yang menggunakan kurir data seperti yang dibahas dalam penelitian ini, informasi yang dipertukarkan merupakan data-data yang utuh. Data langsung diambil menggunakan media penyimpanan pada *source* dan langsung dibawa ke *destination* oleh kurir data dengan menggunakan alat transportasi apapun. Skenario ini mengatasi permasalahan keterbatasan infrastruktur jaringan dan letak geografis yang biasanya ada pada daerah pedesaan bahkan yang sangat terpencil, dimana penyediaan komunikasi data atau Internet kurang memungkinkan.

2.3 Pengolahan Data

Pengolahan data merupakan proses mengatur dan memanipulasi data untuk mendapatkan struktur informasi yang sesuai dengan yang diinginkan. Pada proses manipulasi dan pengaturan data digital, informasi yang disimpan dalam file teks biasa akan sulit untuk diolah dan diproses, sehingga ada solusi lain, yaitu menggunakan sistem basis data.

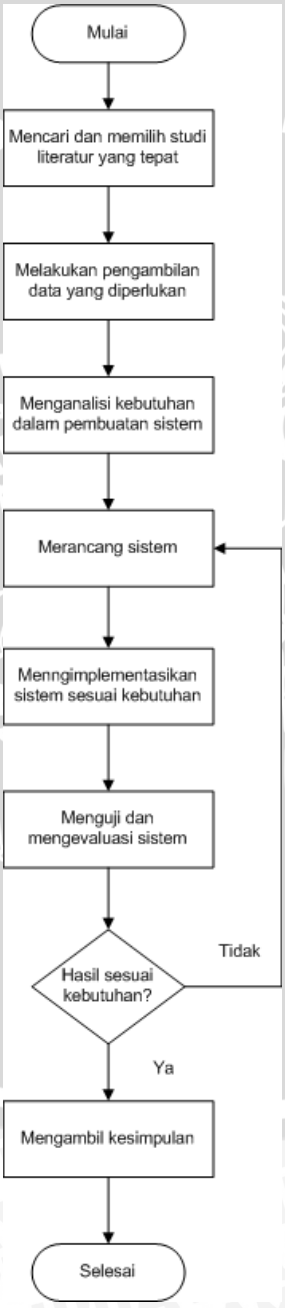
Dalam pengolahan data digital menggunakan basis data diperlukan sebuah perangkat lunak *Database Management System* (DBMS). DBMS mengatur proses manipulasi data seperti *insert, update*, dan *delete* dengan menggunakan perintah-perintah terstruktur, sehingga mudah digunakan. DBMS dapat diintegrasikan dengan aplikasi *web* maupun *desktop*.

MySQL [ASM-13] adalah salah satu perangkat lunak DBMS. MySQL merupakan sistem manajemen basis data yang berbentuk *relational* (RDBMS) yang menggunakan perintah dasar *Structured Query Language* (SQL). Beberapa kelebihan MySQL antara lain:

1. Merupakan *freeware*.
2. Stabil dan tangguh.
3. Fleksibel dengan berbagai pemrograman.
4. Keamanan yang baik.
5. Mendukung transaksi.
6. Perkembangan *software* yang cukup cepat.

BAB III
METODE PENELITIAN

Bab ini menjelaskan langkah-langkah yang digunakan dalam pembuatan sistem berbagi informasi digital di daerah pedesaan. Alur metode penelitian ditunjukkan pada Gambar 3.1



Gambar 3.1. Alur Metode Penelitian

3.1 Studi Literatur

Studi literatur yang digunakan berasal dari buku, artikel ilmiah, *paper*, dan informasi yang tersedia di Internet baik itu dokumentasi proyek maupun artikel. Studi literatur yang dibutuhkan berhubungan dengan hal-hal meliputi:

1. Arsitektur sistem
   - Desain sistem dengan menggunakan diagram UML.
2. Pemrograman berorientasi objek
   - Pemrograman aplikasi berbasis *web* dan aplikasi berbasis *desktop*.
   - Pemrograman berorientasi objek dengan arsitektur Model View Controller (MVC).
3. Pemrograman jaringan
   - Prinsip kerja *Uniform Resource Locator* (URL) dan pengambilan data pada aplikasi yang menggunakan HTTP.
4. Sistem Basis Data
   - Desain dan implementasi basis data.
   - Mekanisme pengolahan data pada basis data.
5. Jaringan komputer dan Internet
   - Prinsip kerja komunikasi data pada jaringan komputer.
   - Prinsip kerja *Delay-Tolerant Network* (DTN).

3.2 Metode Pengambilan Data

Dalam penelitian ini, metode pengambilan data yang digunakan adalah metode observasi. Observasi dilakukan dengan melihat permasalahan dan metode-metode penyediaan infrastruktur berbagi informasi digital di pedesaan yang telah digunakan saat ini. Berdasarkan informasi yang diperoleh, maka dapat ditentukan, kebutuhan perangkat keras, perangkat lunak, dan skenario penggunaan yang sesuai dengan kondisi sosial masyarakat pedesaan.

Berdasarkan hasil observasi, ditemukan bahwa masyarakat pedesaan rata-rata masih rendah dalam literasi mengenai perkembangan teknologi digital saat ini, sehingga sistem ini baik dari segi perangkat keras, perangkat lunak, maupun skenario penggunaannya harus sedapat mungkin mudah dimengerti oleh masyarakat pedesaan.

3.3 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menentukan hal apa saja yang harus diterapkan pada sistem berbagi informasi digital. Daerah pedesaan biasanya memiliki infrastruktur yang terbatas, maka dari itu kebutuhan aplikasi dan kebutuhan lingkungan sistem yang meliputi kebutuhan perangkat keras dan perangkat lunak yang disesuaikan dengan standar minimum.

3.3.1 Kebutuhan Aplikasi

Aplikasi yang dibangun dalam sistem berbagi informasi digital, secara khusus mengacu pada kebutuhan *user* di pedesaan yang ingin mendapatkan informasi digital. Kebutuhan aplikasi ini meliputi:

- Aplikasi harus mampu menyediakan informasi digital bagi para *user* di pedesaan sesuai dengan yang diinginkannya.
- Aplikasi harus memiliki antarmuka pengguna yang sederhana, sehingga mudah dimengerti penggunaannya oleh *user* di pedesaan.
- Aplikasi harus mampu menyediakan informasi yang belum tersedia sesuai dengan keinginan *user* di pedesaan.

3.3.2 Kebutuhan Lingkungan Sistem

- **Kebutuhan Perangkat Keras**

  Kebutuhan perangkat keras dalam penelitian ini meliputi semua perangkat yang harus ada dalam sistem berbagi informasi digital diantaranya:

  - Satu unit PC/Laptop atau *end device* lain yang memiliki *browser* untuk akses informasi digital. Perangkat ini digunakan oleh *user*.
  - Satu unit USB Flash Drive (FAT32, NTFS) sebagai media pertukaran data. Perangkat ini digunakan oleh kurir data.
  - Satu unit PC/Laptop (x86 family of processors) sebagai server yang diakses oleh *user* dan juga kurir data.

- **Kebutuhan Perangkat Lunak**

  Kebutuhan perangkat lunak dalam penelitian ini adalah beberapa aplikasi yang akan diinstal pada server diantaranya:

  - Apache HTTP Server sebagai penyedia *web service*.
  - Modul bahasa pemrograman PHP versi 5.3.1.
  - DBMS MySQL versi 5.1.41 untuk penyimpanan data.
  - Java Runtime Environtment Version 1.7. 0_25 sebagai lingkungan sistem aplikasi Java.

3.4 Perancangan Sistem

Sistem berbagi informasi digital dirancang setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Perancangan sistem ini meliputi:

- **Desain Arsitektur Sistem**

  Merancang skenario keseluruhan sistem yang dibangun dalam beberapa tahap meliputi:

  - Mendefinisikan bagian-bagian yang terdapat pada sistem berbagi informasi digital untuk daerah terpencil.
  - Mendefinisikan alur kerja seorang *user* untuk mendapatkan informasi digital.
  - Mendefinisikan langkah-langkah seorang kurir data sebagai pengganti mekanisme *store and forward* pada DTN dalam mencari informasi digital sesuai permintaan *user* tanpa adanya koneksi jaringan.

- **Fitur Aplikasi**

  Mendefinisikan hal-hal yang dapat dilakukan oleh aplikasi yang merupakan bagian dari sistem. Fitur aplikasi dirancang berdasarkan kebutuhan fungsional dan non-fungsional.

- **Desain Aplikasi**

  Merancang arsitektur penyusun aplikasi dan menyediakan langkah-langkah penggunaan aplikasi yang terdiri dari perancangan UML.

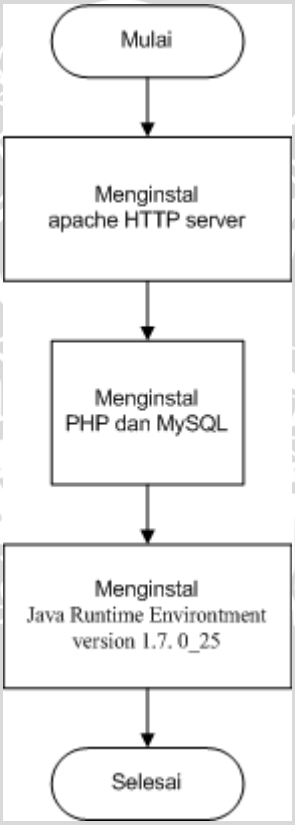
  - Use Case Diagram.
  - Class Diagram.

- Desain Basis Data.
- Sequence Diagram.
- Activity Diagram

3.5 Implementasi

Implementasi dilakukan dengan membangun aplikasi pada sistem yang sebelumnya telah dirancang. Langkah-langkahnya meliputi:

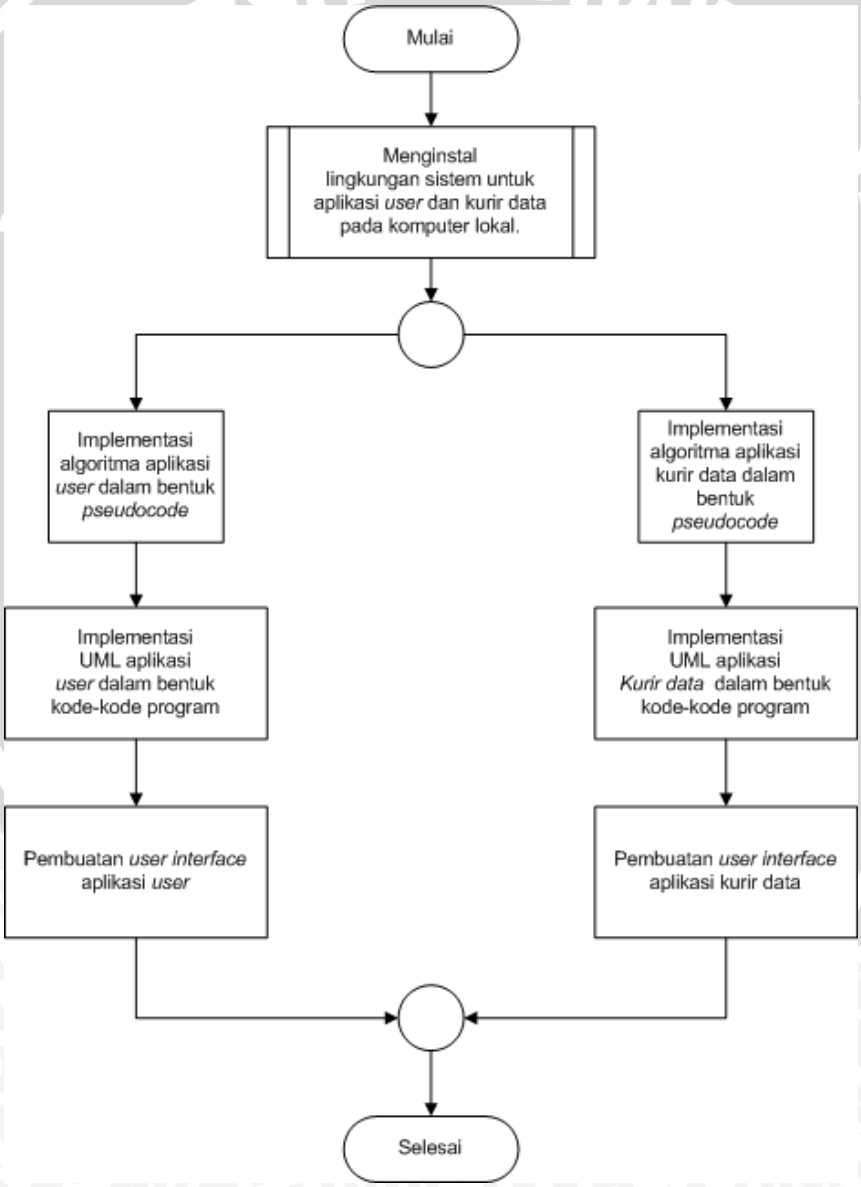
- Menginstal Apache HTTP Server, PHP, DBMS MySQL, dan Java Runtime Environtment Version 1.7. 0_25. pada PC/Laptop server seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3.2. Alur Implementasi Lingkungan Sistem

- Implementasi aplikasi meliputi:
  1. Implementasi setiap algoritma yang ditampilkan dalam bentuk *pseudocode* dan hubungannya dengan algoritma lain.
  2. Implementasi UML ke dalam kode-kode program yang berakhir pada pembuatan antarmuka pengguna.

Gambar 3.3 menunjukkan alur implementasi aplikasi sistem secara keseluruhan, dimulai dari implementasi lingkungan sistem dan selanjutnya implementasi aplikasi.



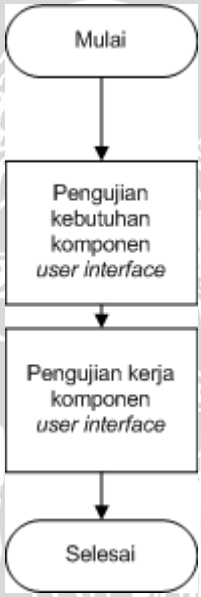
Gambar 3.3. Alur Implementasi Aplikasi Sistem

3.6 Pengujian dan Analisis

Pengujian sistem dilakukan agar dapat diketahui apakah sistem sudah bekerja sesuai dengan spesifikasi yang telah ditentukan serta diharapkan menemukan kesalahan-kesalahan (*bug*) pada aplikasi. Pengujian ini meliputi beberapa pengujian perangkat lunak diantaranya:

3.6.1 Pengujian Kebutuhan Fungsional

Pengujian kebutuhan fungsional dalam penelitian ini meng-gunakan *Black Box Testing*, yaitu tes yang berbasis *requirements* dan *functionality*. Pengujian ini dilakukan dengan menginspeksi kelengkapan dan fungsi kerja komponen antarmuka pengguna. Gambar 3.4 menunjukkan alur pengujian fungsional aplikasi *user* dan kurir data.

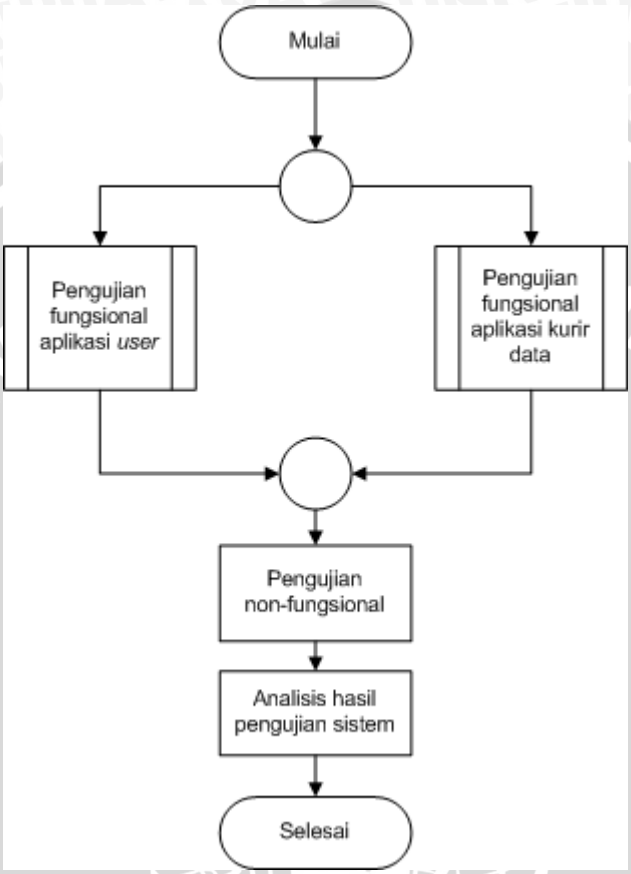


Gambar 3.4. Alur Pengujian Fungsional Aplikasi *User* dan Kurir Data

3.6.2 Pengujian Kebutuhan Non-Fungsional

Pengujian terhadap sistem yang tidak berhubungan dengan fungsi spesifik sistem. Pengujian ini secara khusus dilakukan untuk menjamin sistem berbagi informasi digital memiliki keandalan saat terjadi hal-hal yang tidak terduga pada lingkungan sistem pada saat proses implementasi.

Setelah sistem diuji, dilakukan analisis untuk mengetahui tingkah laku sistem sebagai acuan kesesuaian implementasi dan pengujian terhadap kebutuhan sistem. Gambar 3.5 menunjukkan alur pengujian dan analisis sistem yang dilakukan secara keseluruhan.



Gambar 3.5. Alur Pengujian dan Analisis Sistem

3.7 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem berbagi informasi digital yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta memberikan pertimbangan atas pengembangan sistem selanjutnya.

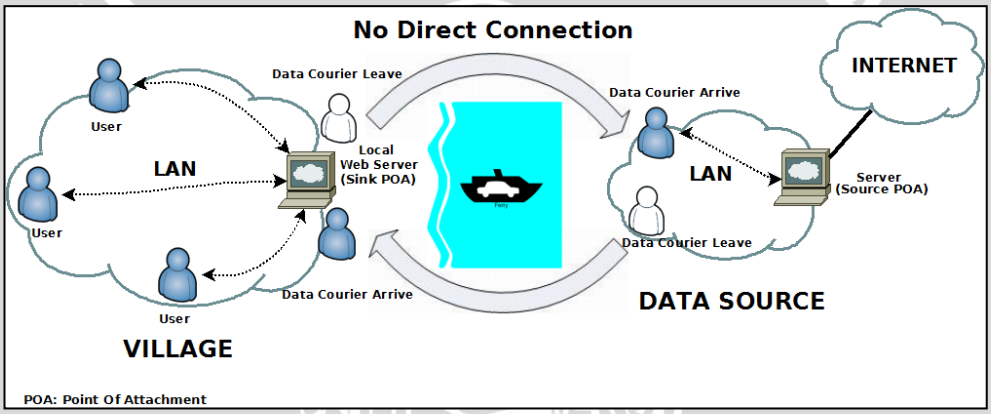
BAB IV
PERANCANGAN

4.1 Perancangan

Berikut adalah perancangan sistem yang akan dibangun, terdiri dari desain arsitektur sistem, fitur aplikasi, dan desain aplikasi.

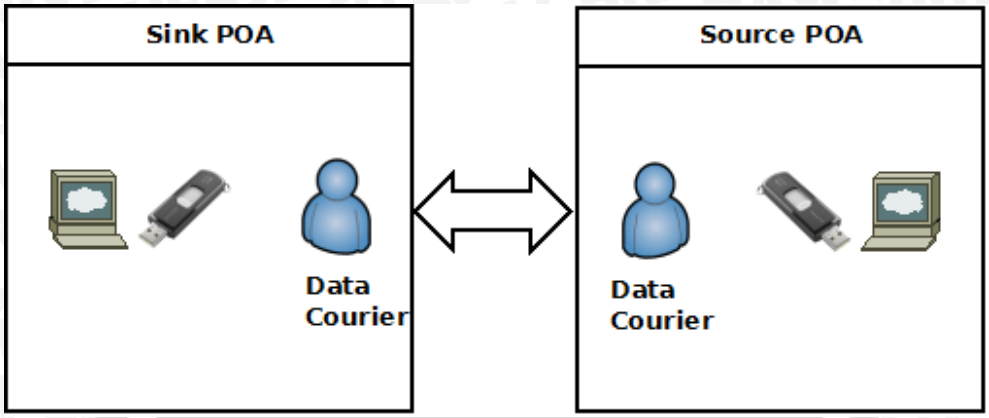
4.1.1 Desain Arsitektur Sistem

Skenario perancangan pada Gambar 4.1 menunjukkan bahwa tidak ada koneksi jaringan secara langsung antara desa dan sumber data, sehingga proses pengiriman dan penerimaan informasi digital (file-file) dijembatani oleh seorang kurir data secara berkesinambungan.



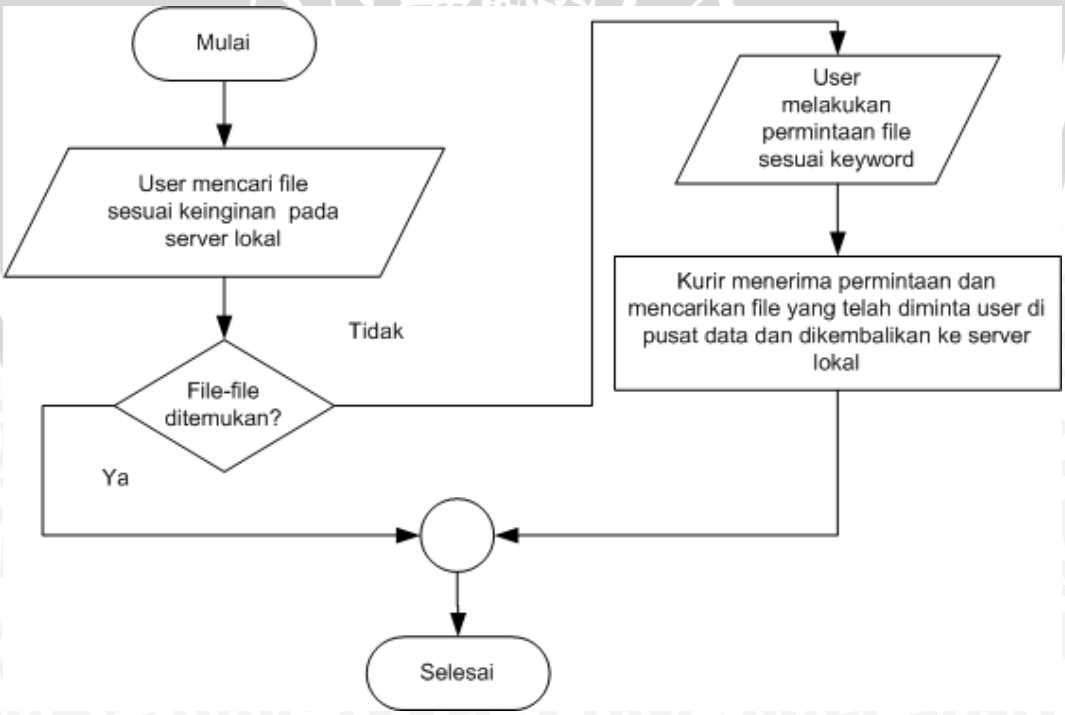
Gambar 4.1. Skenario Sistem

Desa memiliki sebuah sistem yang menyediakan informasi-informasi yang diperlukan para *user* dengan menyediakan cara akses layaknya browsing di Internet. Sistem informasi yang diakses oleh *user* berada dalam *Local Area Network* (LAN) pada suatu tempat, misalnya balai desa. Seorang *user* dapat melakukan pencarian informasi digital dalam sistem ini dan apabila tidak menemukan informasi yang diinginkannya, maka *user* dapat melakukan permintaan sesuai keinginannya pada *local web server* dalam LAN tersebut. Permintaan-permintaan tersebut akan disimpan sementara oleh sistem sampai ada kurir data yang berangkat ke sumber data dengan alat transportasi. Sumber data dapat merupakan sebuah kota maupun tempat lain yang dianggap memiliki informasi yang cukup lengkap.



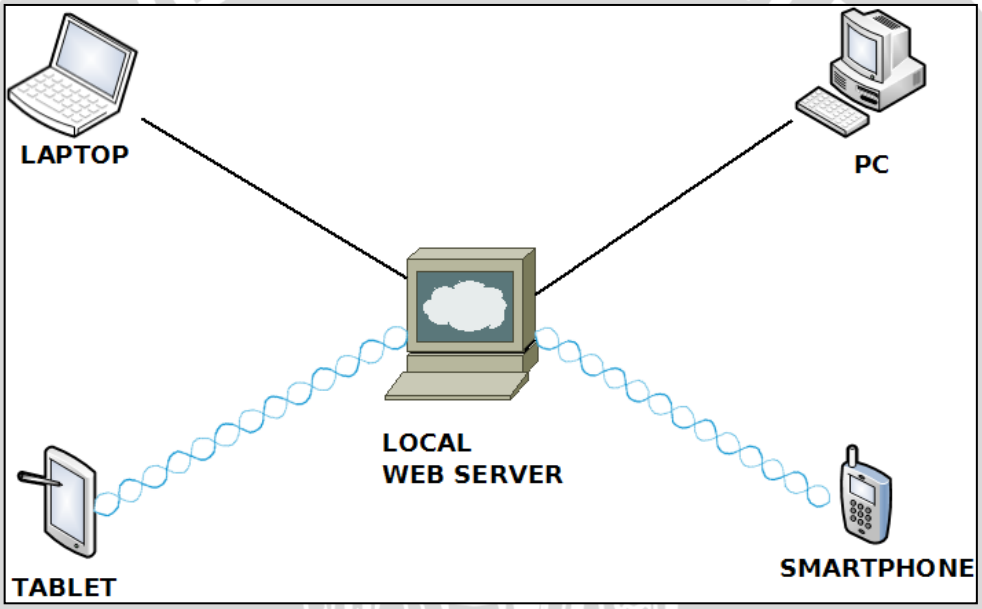
Gambar 4.2. Sistem Kerja Kurir Data

Pada Gambar 4.1 terdapat titik akses yang diistilahkan dengan *Point of Attachment* (POA). Titik akses yang pertama, yaitu *local web server* disebut dengan Sink POA dan titik akses yang kedua adalah server yang ada di sumber data disebut dengan Source POA. Gambar 4.2 menunjukkan bahwa seorang kurir data memiliki dua peran. Peran pertama, yaitu mengakses Sink POA dan peran kedua, yaitu mengakses Source POA.



Gambar 4.3. Alur Skenario Sistem

Berdasarkan alur skenario sistem yang ditunjukkan oleh Gambar 4.3, dengan berbekal file-file permintaan dari Sink POA, kurir data yang menuju server di sumber data akan mengambil informasi-informasi sesuai file-file permintaan tersebut di Source POA. Apabila informasi tidak ditemukan di *harddisk* server, kurir data dapat melakukan pencarian di Internet (bila ada koneksi Internet) dan kemudian membawa hasil pencarian informasi tersebut ke *local web server* di desa. Ketika hasil pencarian sudah sampai di Sink POA, maka *user* dapat melakukan pencarian data kembali berdasarkan permintaan sebelumnya.



Gambar 4.4. Arsitektur LAN pada Sink POA

Gambar 4.4 menjelaskan bahwa aplikasi pada Sink POA dapat diakses oleh semua jenis *end device* yang memiliki *browser*. Tidak ada batasan terhadap *end user*, siapapun dapat mengakses informasi yang ada.

4.1.2 Fitur aplikasi

Ada dua buah aplikasi yang dibangun, yaitu File Browser (yang digunakan oleh *user*) dan File Collector (yang digunakan oleh kurir data).

4.1.2.1 Kebutuhan Fungsional

- File Browser (*Web Based Application*)

  File Browser adalah sebuah aplikasi berbasis web dengan tampilan sederhana yang menyediakan fitur pencarian informasi digital sesuai keinginan pengguna. Beberapa tugas yang harus dapat dilakukan oleh File Browser diantaranya:

  1. Aplikasi ini harus dapat menampilkan nama seluruh file PDF yang berisi konten sesuai dengan kata kunci masukan dari pengguna.
  2. Menampilkan isi file secara keseluruhan ketika pengguna ingin membaca salah satu hasil pencarian.
  3. Menyediakan kolom permintaan file sesuai kata kunci bila file yang dicari tidak ada atau pengguna merasa hasil yang ada kurang memuaskan.
  4. Melakukan validasi terhadap kata kunci masukan dari pengguna untuk menjamin tidak ada kata yang tidak sesuai dengan kosa kata dalam Bahasa Indonesia atau Bahasa Inggris. Selain itu, aplikasi juga harus dapat mengidentifikasi apabila permintaan dari pengguna sudah ada sebelumnya.

- File Collector (*Desktop Application*)

  File Collector adalah sebuah aplikasi *desktop* dengan tampilan sederhana yang menyediakan fitur penyalinan/pengunduhan informasi digital yang sebelumnya diminta oleh pengguna File Browser. Aplikasi ini digunakan oleh kurir data ketika berada di Sink POA dan di Source POA. Aplikasi ini memiliki fungsi terpisah. Beberapa tugas yang harus dapat dilakukan oleh File Collector diantaranya:

  - Sink POA
    1. Ketika berada di Sink POA, maka aplikasi harus dapat menerima semua permintaan dalam format XML yang diminta

oleh pengguna File Browser dan menyimpannya pada sebuah folder khusus.

2. Memindahkan semua file PDF yang sebelumnya sudah didapatkan dari sumber data ke File Browser.
3. Meng-*update* basis data File Browser, sehingga seluruh file yang sudah diterima dapat diakses oleh pengguna File Browser.
4. Memberitahukan dan menawarkan solusi kepada kurir data ketika ada file yang tidak dapat dioperasikan karena telah dibuka oleh aplikasi lain.
5. Menghapus semua permintaan bila file sudah didapatkan.

- Source POA

1. Ketika berada di Source POA, maka aplikasi ini harus dapat melakukan pencarian file-file PDF pada satu atau lebih direktori pada server dan menyalinnya ke dalam folder khusus. File-file PDF yang disalin adalah file-file yang berisi informasi yang diminta oleh pengguna File Browser di Sink POA.
2. Mengunduh file dari Internet bila diperlukan. Mekanismenya sama dengan saat penyalinan.
3. Memberitahukan dan menawarkan solusi kepada kurir data ketika USB Flash Drive mengalami kekurangan *free space memory* saat terjadi operasi file.

4.1.2.2 Kebutuhan Non-Fungsional

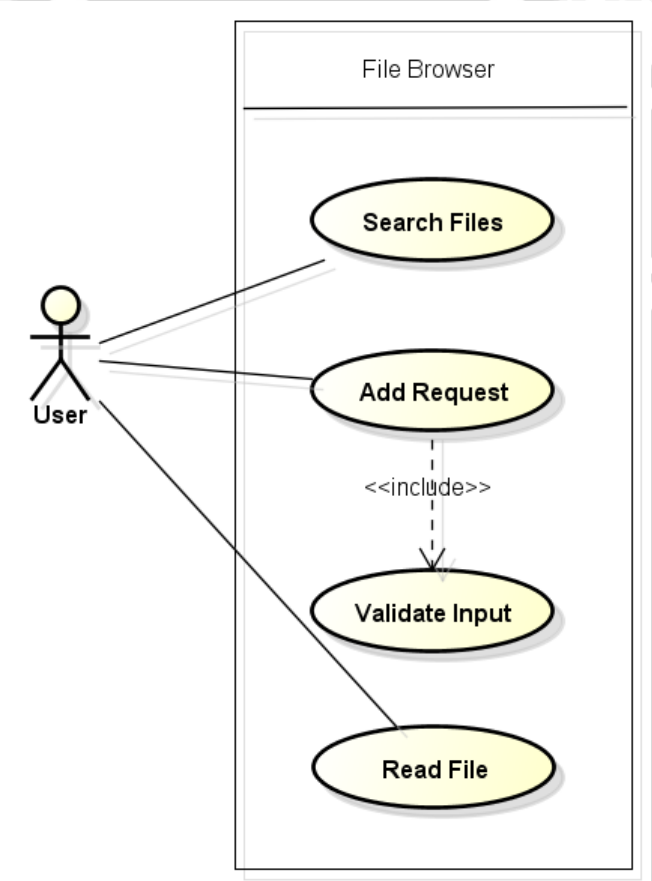
1. Sistem harus dapat menangani kondisi ketika listrik mati secara mendadak, mengingat di daerah pedesaan sering terjadi pemadaman listrik.
2. Sistem harus dapat menangani kondisi ketika misalkan USB Flash Drive yang dibawa oleh kurir data hilang atau terjadi sesuatu yang menyebabkan data di USB Flash Drive tersebut tidak dapat diakses. Sistem harus menjamin bahwa data selalu memiliki *backup* dan menghindari penghapusan yang beresiko untuk menjamin keandalan sistem.

4.1.3 Desain Aplikasi

4.1.3.1 Use Case Diagram

Use Case Diagram menunjukkan apa saja yang dapat dilakukan oleh *actor* (dalam hal ini, yaitu *user* dan kurir data) pada masing-masing aplikasi yang digunakannya secara umum.

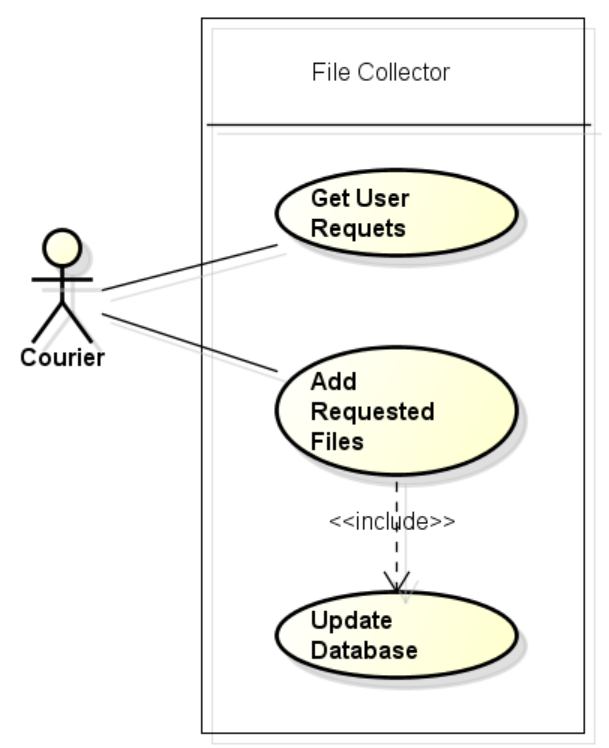
- ***User***



Gambar 4.5. Use Case: *User*

Gambar 4.5 menunjukkan bahwa *user* memiliki tiga aksi, yaitu melakukan pencarian file, menambah permintaan, dan membaca file.

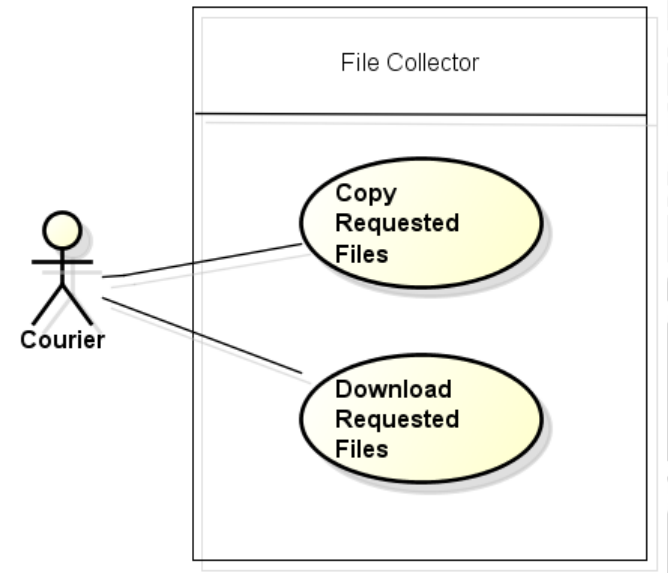
- **Kurir Data di Sink POA**



Gambar 4.6. Use Case: Kurir Data di Sink POA

Gambar 4.6 menunjukkan bahwa seorang kurir data di Sink POA memiliki tiga aksi, yaitu menerima permintaan *user*, menambah file-file permintaan, dan meng-*update* basis data.

- **Kurir Data di Source POA**



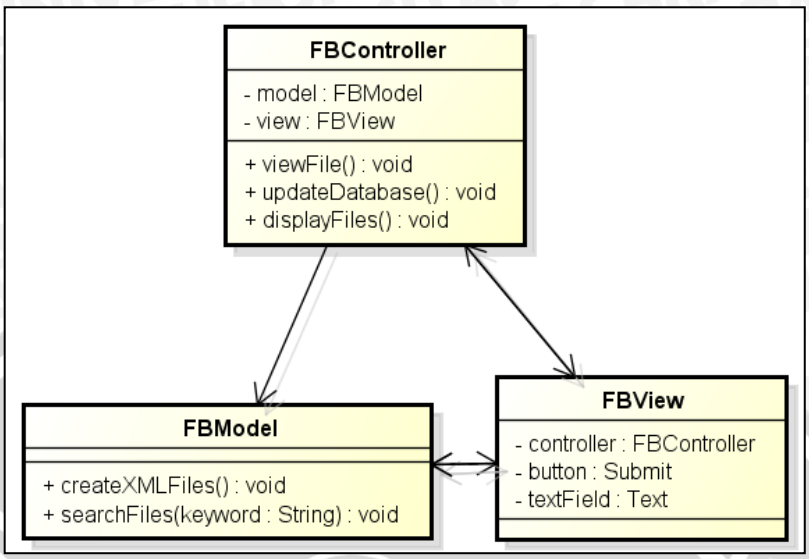
Gambar 4.7. Use Case: Kurir Data di Source POA

Gambar 4.7 menunjukkan bahwa seorang kurir data di Source POA memiliki dua aksi, yaitu melakukan penyalinan dan pengunduhan file sesuai permintaan.

4.1.3.2 Class Diagram

Pola perancangan *Class Diagram* menggunakan arsitektur MVC. MVC memisahkan tiap komponen aplikasi berdasarkan tampilan (View), kelakuan (Model), dan pengontrol keduanya (Controller).

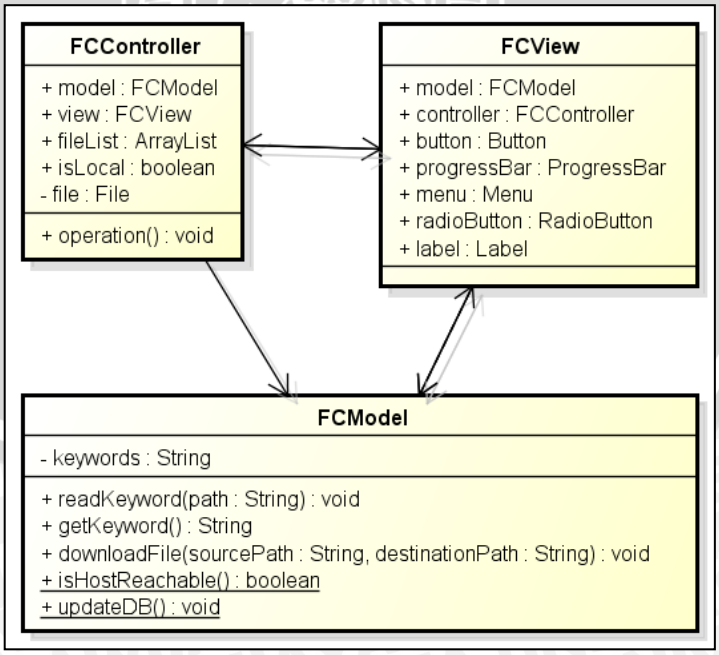
- **File Browser**



Gambar 4.8. Pola MVC pada File Browser

Gambar 4.8 merupakan relasi antara FBModel, FBView, dan FBController. FBController mengatur interaksi yang terjadi antara FBView dan FBModel.

- **File Collector**



Gambar 4.9. Pola MVC pada File Collector

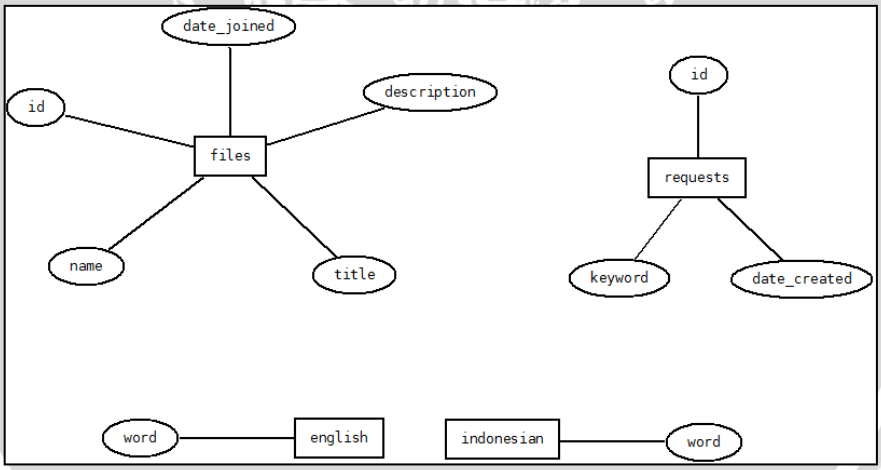
Gambar 4.9 merupakan relasi antara FCModel, FCView, dan FCController. FCController mengatur interaksi yang terjadi antara FCView dan FCModel.

- **Perancangan Basis Data**

Perancangan basis data hanya dilakukan pada aplikasi File Browser.

Entity Diagram

Entitas yang ada terdiri dari entitas *files*, *requests*, *english*, dan *indonesian* seperti yang ditunjukkan Gambar 4.10. Entitas *files* berisi informasi berupa *metadata* dari file-file yang disimpan di server dan entitas *requests* menampung sejumlah kata kunci yang didefinisikan oleh pengguna sistem, sedangkan entitas *english* dan *indonesian* adalah daftar kata-kata dalam Bahasa Inggris dan Bahasa Indonesia yang dapat dimasukkan sebagai kata kunci (*keyword*).



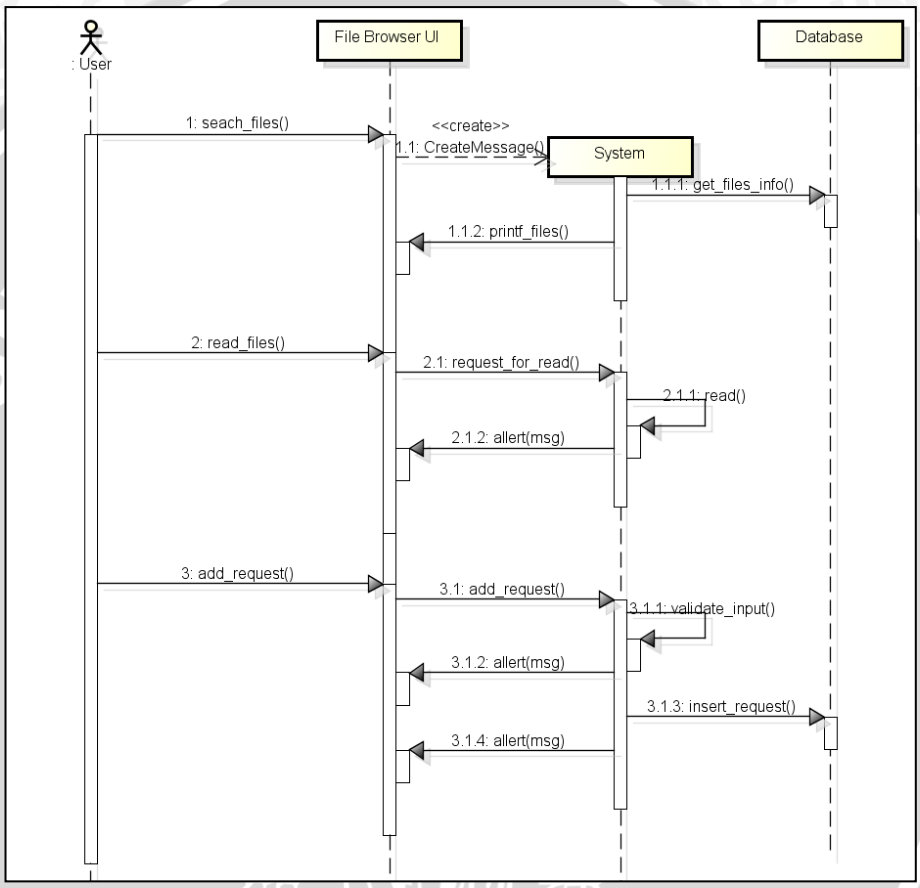
Gambar 4.10. ER Diagram Basis Data File Browser

Pada diagram entitas tersebut, tidak ada relasi antar entitas. Tiap entitas berdiri sendiri karena tidak ada keterkaitan.

4.1.3.3 Sequence Diagram

Sequence Diagram menunjukan apa saja aksi yang dapat dilakukan aktor pada suatu fungsi secara berurutan pada tiap aplikasi. Diagram ini hanya menampilkan urutan aksi per fungsi dan tidak menunjukkan kaitannya dengan fungsi lain.

- ***User***



Gambar 4.11. Sequence Diagram: *User*

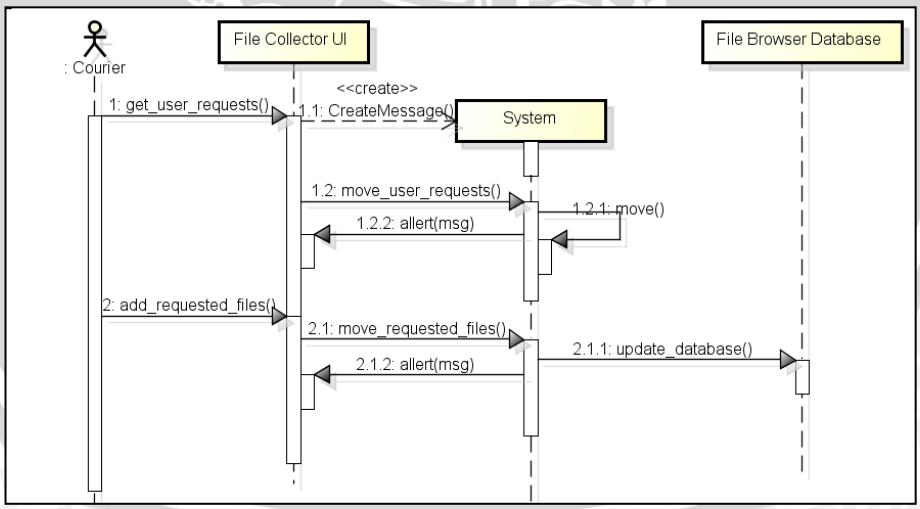
Sequence Diagram untuk *user* pada Gambar 4.11 menunjukkan tiga hal yang dapat dilakukan, yaitu *search_file()*, *read_file()*, dan *add_request()*. Berikut ini adalah penjelasan masing-masing interaksi yang dilakukan.

- ***search_file()***
  - *User* meminta sistem untuk menampilkan file-file sesuai keinginannya.

- Sistem menampilkan daftar file-file dalam basis data sesuai kata kunci masukan *user*.

- ***read_file()***
  - Sistem menampilkan file-file sesuai kata kunci yang dimasukkan oleh *user*.
  - *User* memilih salah satu file.
  - Sistem membuka file yang telah dipilih oleh *user*.
  - Sistem menampilkan pesan *error* bila terjadi kesalahan.
- ***add_request()***
  - *User* menambahkan kata kunci permintaan kepada sistem.
  - Sistem melakukan validasi kata kunci masukan.
  - Sistem menampilkan pesan *error* bila masukan tidak valid.
  - Apabila masukan valid, sistem akan menyimpan kata kunci ke basis data.
  - Sistem menampilkan pesan *error* bila terjadi kesalahan.

➢ **Kurir Data di Sink POA**

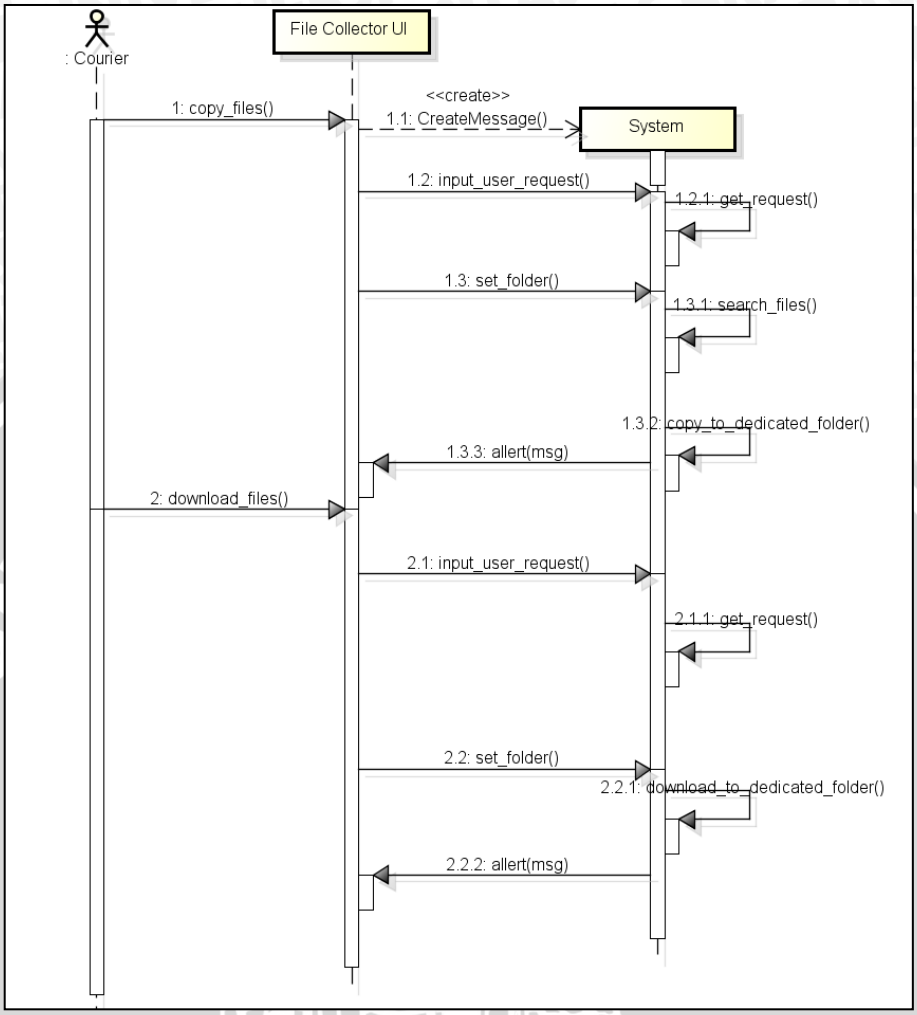


Gambar 4.12. Sequence Diagram: Kurir Data di Sink POA

Sequence Diagram untuk kurir data di Sink POA pada Gambar 4.12 menunjukkan dua hal yang dapat dilakukan oleh kurir data, yaitu *get_user_requests()*, dan *add_requested_files()*. Berikut ini adalah penjelasan masing-masing interaksi yang dilakukan.

- ***get_user_requests()***
  - Kurir data meminta sistem untuk memindahkan file-file permintaan dari *local web server* ke dalam folder khusus yang ditentukan oleh program yang dibawa kurir data.
  - Sistem akan menampilkan pesan *error* bila terjadi kesalahan.
- ***add_requested_file()***
  - Kurir data meminta sistem untuk memindahkan semua file hasil pencarian yang ada dalam media penyimpanan ke *local web server*.
  - Sistem memindahkan file-file dan melakukan pembaharuan terhadap basis data.
  - Sistem menampilkan *error* bila terjadi kesalahan.

- **Kurir Data di Source POA**



Gambar 4.13. Sequence Diagram: Kurir Data di Source POA

Sequence Diagram untuk kurir data di Source POA pada Gambar 4.13 menunjukkan dua hal yang dapat dilakukan, yaitu *copy_files()*, dan *download_files()*. Berikut ini adalah penjelasan masing-masing interaksi yang dilakukan.

- ***copy_files()***
  - Kurir data meminta sistem mengambil semua informasi pada file-file permintaan.
  - Sistem mengumpulkan semua permintaan dan menentukan folder tujuan penyalinan.

- Sistem melakukan pencarian file-file sesuai kumpulan permintaan.
- Sistem menyalin file-file sesuai permintaan ke dalam folder.
- Sistem akan menampilkan pesan *error* bila terjadi kesalahan.

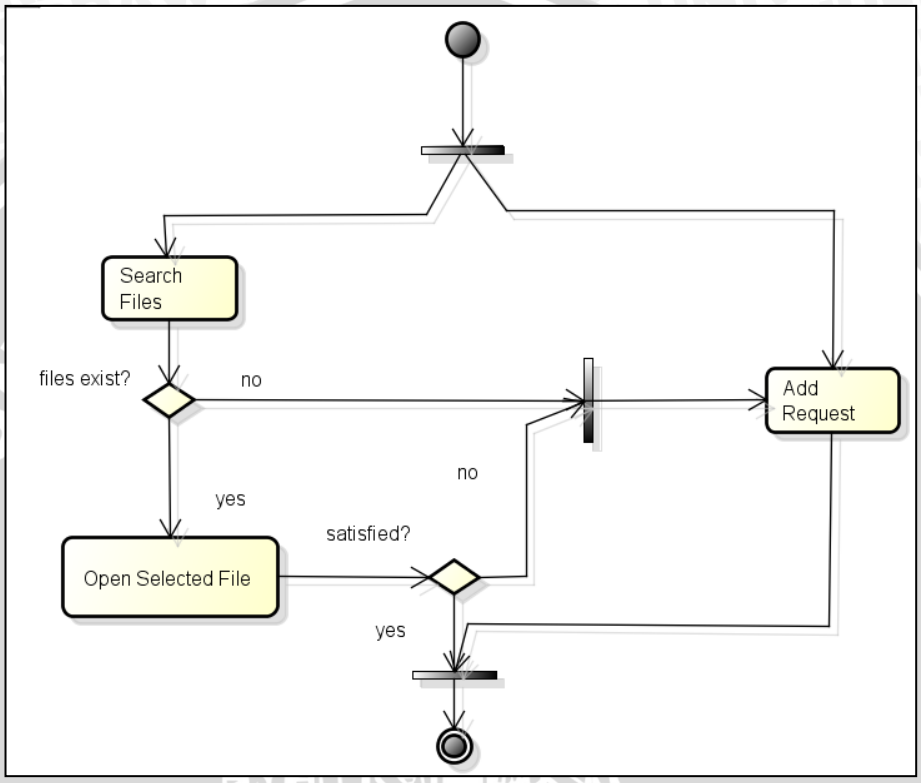
➢ ***download_files()***

- Kurir data meminta sistem mengambil semua informasi pada file-file permintaan.
- Sistem mengumpulkan semua permintaan dan menentukan folder tujuan penyalinan.
- Sistem melakukan pencarian file-file sesuai kumpulan permintaan.
- Sistem mengunduh file-file sesuai permintaan ke dalam folder.
- Sistem akan menampilkan pesan *error* bila terjadi kesalahan.

4.1.3.4 Activity Diagram

Activity Diagram menunjukan alur penggunaan aplikasi, atau dengan kata lain, bagaimana tingkah laku sistem ketika suatu aksi diberikan sebagai masukan.

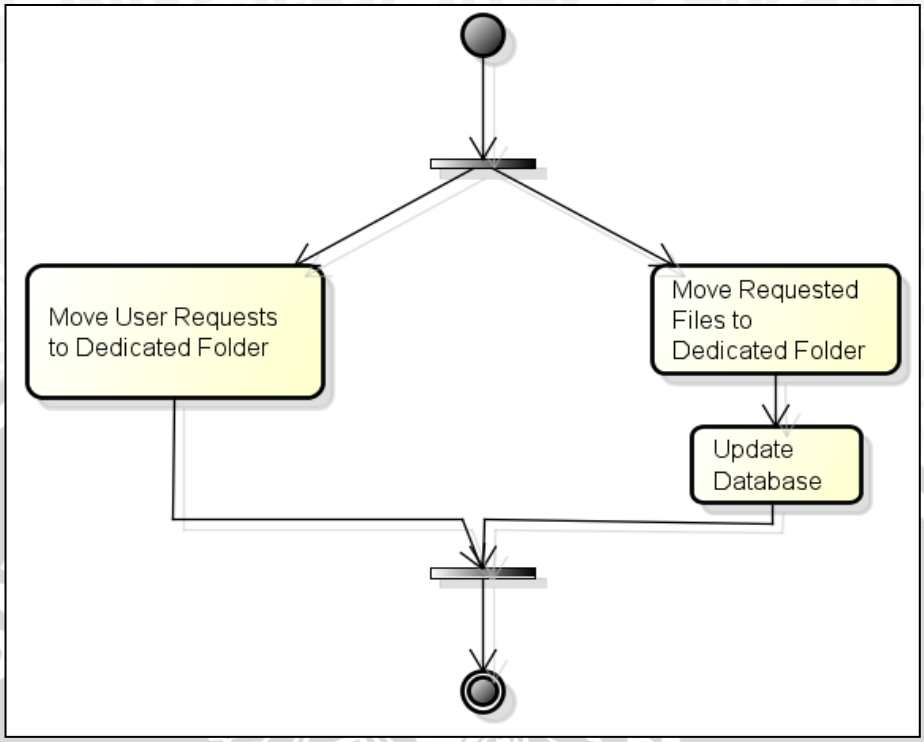
- ***User***



Gambar 4.14. Activity Diagram: *User*

Gambar 4.14 menunjukkan bahwa ketika *user* memasukkan kata kunci untuk melakukan pencarian file, sistem akan memeriksa apakah kata kunci cocok dengan isi file-file yang tersedia. Bila ada, maka file-file yang cocok ditampilkan, kemudian *user* dapat membuka file sesuai pilihan. Apabila tidak ada file yang ditemukan atau hasil kurang memuaskan, maka *user* dapat melakukan permintaan kepada sistem.

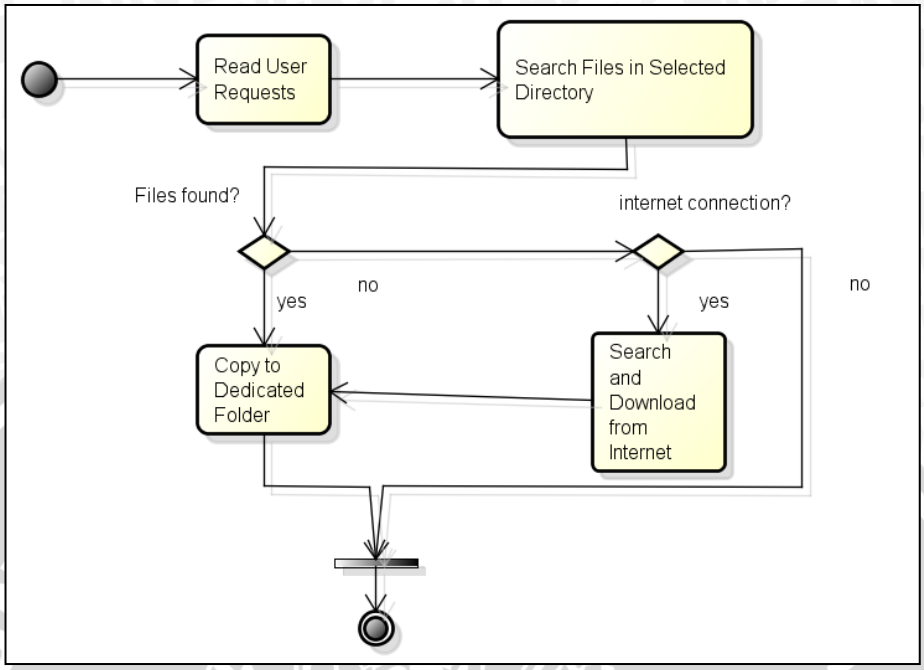
- **Kurir Data di Sink POA**



Gambar 4.15. Activity Diagram: Kurir Data di Sink POA

Gambar 4.15 menunjukkan bahwa ketika berada di Sink POA, kurir data dapat melakukan 2 hal. Pertama, memindahkan permintaan *user* yang sudah tersedia dalam bentuk file XML ke sebuah folder khusus dalam media penyimpanan. Kedua, memindahkan file-file yang sudah didapatkan sebelumnya di sumber data ke sebuah folder khusus pada server lokal, kemudian program secara otomatis akan meng-*update* basis data server.

- **Kurir Data di Source POA**



Gambar 4.16. Activity Diagram: Kurir Data di Source POA

Gambar 4.16 menunjukkan bahwa ketika berada di Source POA, kurir data menjalankan program yang melakukan pembacaan terhadap file-file permintaan dari *user* sebelumnya. Berdasarkan informasi yang tersedia, program melakukan pencarian file-file pada sebuah folder yang dipilih oleh kurir data. Apabila file ditemukan, maka file tersebut akan disalin ke sebuah folder khusus. Apabila file tidak ditemukan, maka kurir data dapat menentukan apakah file akan dicari di internet atau tidak. Ketika file yang dicari di internet ditemukan, maka program juga akan mengunduh file tersebut dan dipindahkan ke folder khusus.

4.2 Implementasi

Berikut adalah implementasi sistem yang dibangun. Bagian-bagian implementasi ini terdiri dari lingkungan implementasi, batasan implementasi, implementasi metode, dan implementasi antarmuka.

4.2.1 Lingkungan Implementasi

- **File Browser**

  Aplikasi File Browser dalam implementasinya dibangun dengan menggunakan bahasa pemrograman PHP versi 5.3.1. Untuk mendapatkan struktur sistem yang baik, aplikasi ini dibangun di dalam Code Igniter yang merupakan framework PHP. Code Igniter yang digunakan adalah Code Igniter versi 2.1.0. Data yang diproses disimpan dalam basis data menggunakan MySQL versi 5.1.41. Lingkungan ini dijalankan pada Apache HTTP Server. Baik PHP, MySQL, dan Apache HTTP Server sudah satu paket dengan XAMPP versi 1.7.3.

- **File Collector**

  Aplikasi File Collector dibangun dengan menggunakan bahasa pemrograman Java. Aplikasi File Collector harus ditempatkan pada USB Drive karena sistem dibuat agar hanya mengenali USB Drive. Aplikasi ini dapat diakses secara *portable*, sehingga tidak perlu ada proses instalasi.

4.2.2 Batasan Implementasi

Sistem Operasi yang menggunakan aplikasi File Collector harus sudah terinstalasi Java Runtime Environtment (JRE) versi 1.7.0_25 karena ada beberapa fungsi yang menggunakan Java Development Kit (JDK) versi 1.7.0_12 yang tidak bekerja di versi JRE yang lebih lama.

4.2.3 Implementasi Metode

Pada bagian ini ditampilkan algoritma-algoritma pada tiap aplikasi dalam bentuk *pseudocode.* Algoritma-algoritma yang ditampilkan adalah algoritma yang menjadi fungsi-fungsi inti dari aplikasi. Pada bagian ini juga dijelaskan hubungan antar algoritma-algoritma tersebut.

- **File Browser**

```
prosedur insert_files ()
{ memasukkan informasi file ke basis data }

Deklarasi
    file_dir    : string
    pdf         : pdf_model
    pdf_info    : pdf_info_model

Algoritma
  foreach← 0 to dir_length do
    if(file is file and file is pdf)
        save file name to array
        decode pdf by pdf_model and save to array
        extract pdf metadata by pdf_info_model and
    save to array
        if(!database contain file)
            insert file data into database
        endif
    endif
  endforeach
endprosedur
```

Gambar 4.17. Prosedur insert_files()

Prosedur *insert_files()* pada Gambar 4.17 digunakan untuk memasukkan semua informasi yang ada pada tiap file ke dalam basis data. Hal ini dilakukan agar dapat dilakukan pencarian file dengan mudah oleh sistem. Cara kerjanya, yaitu file-file PDF baru yang telah dipindahkan ke sebuah direktori tertentu diekstrak isi dan *metadata*-nya. Isi file dicocokkan dengan semua isi file yang ada di dalam basis data sebelumnya untuk memeriksa apakah file tersebut sudah ada untuk

menghindari duplikasi. Pencocokan isi file dilakukan karena isi file adalah unik. Apabila file belum ada, maka informasi-informasi tersebut dimasukkan ke dalam basis data.

```
prosedur delete_files ()
{ sinkronisasi jumlah antara file di basis data dan folder khusus }

Deklarasi
    file_dir    : string

Algoritma
  foreach← 0 to dir_length do
    if(file is file and file is pdf)
            save dir file name to array
            save database file name to array
            compare arrays and move diference to a new array
    for← 0 to new_array do
            delete files in database
    endfor
    endif
  endforeach
endprosedur
```

Gambar 4.18. Prosedur delete_files()

Prosedur *delete_files()* pada Gambar 4.18 digunakan untuk mensinkronisasi seluruh informasi yang ada di dalam basis data dengan sebuah folder dimana file-file tersebut disimpan. Hal ini dilakukan mengingat jumlah file yang ada dalam basis data belum tentu sama

dengan yang ada di folder untuk menghindari kesalahan saat pembacaan file. Objek sinkronisasi adalah nama file karena nama file adalah unik.

```
prosedur insert_request ()
{ memasukkan data permintaan ke basis data }

Deklarasi
    keyword_list      : array

Algoritma
keyword_list ← explode(keyword)
  for← 0 to keyword_list _length do
        if(keyword_list match vocabullary)
          if(!database contain file)
                    insert keyword into database
          endif
        endif
  endfor
endprosedur
```

Gambar 4.19. Prosedur insert_request()

Prosedur *insert_request()* pada Gambar 4.19 digunakan untuk memasukkan permintaan para pengguna ke dalam basis data untuk selanjutnya diubah ke bentuk file XML. Kata kunci permintaan harus sesuai dengan kosa kata Bahasa Inggris dan Bahasa Indonesia yang didefinisikan di dalam basis data. Apabila tidak sesuai, maka hal tersebut akan dianggap tidak valid. Sistem juga memeriksa apakah kata kunci tersebut sudah ada dalam basis data untuk menghindari duplikasi.

```
prosedur delete_request ()
{ sinkronisasi jumlah antara permintaan di basis data dan folder khusus }

Deklarasi
    file_dir    : string

Algoritma
  foreach← 0 to dir_length do
    if(file is file and file is xml)
            save dir file name to array
            save database request keyword to array
            compare arrays and move diference to a new array
    for← 0 to new_array do
            delete request in database
    endfor
    endif
  endforeach
endprosedur
```

Gambar 4.20. Prosedur delete_request()

Prosedur *delete_request()* pada Gambar 4.20 digunakan untuk melakukan sinkronisasi antara kata kunci di dalam basis data dengan yang ada pada sebuah folder khusus. Hal ini dilakukan mengingat adanya kemungkinan tidak semua permintaan *user* didapatkan pada sumber data. Hanya kata kunci yang file-filenya sudah ditemukan yang akan dihapus, sementara kata kunci yang file-filenya belum ditemukan

akan tetap disimpan untuk selanjutnya dilakukan pencarian kembali pada waktu yang akan datang.

- **File Collector**

```
prosedur search(input file: File)
{ mencari file pada sebuah directory }

Deklarasi
    keywords: String[]

Algoritma (rekursif)
if(file isDirectory and canRead)
    for file ← 0 to jumlah_file do
    {apabila direktori, search pada direktori tsb}
        if(file isDirectory and canRead)
            search(file)
        endif
        {apabila merupakan file}
        else
            for file ← 0 to keywords_length do
                if(filename contains keyword)
                    add file to result
                endif
            endfor
        endelse
    endfor
  endif
endprosedur
```

Gambar 4.21. Prosedur search()

Prosedur *search()* pada Gambar 4.21 digunakan untuk melakukan pencarian file-file pada sebuah folder. Proses pencarian dilakukan berdasarkan kata kunci yang sudah disimpan sebelumnya. Apabila isi file mengandung kata kunci, maka nama file tersebut dipindahkan ke *array* untuk selanjutnya disalin oleh program dengan prosedur *copy_file()* yang ditunjukkan pada Gambar 4.23. Prosedur *search()* menggunakan algoritma rekursif *Deep First Search* (DFS) untuk menjamin semua file dalam folder tempat penyimpanan file ditelusuri.

```
prosedur google_search()
{ mencari file di pencarian google }
Deklarasi
    listModel : ListModel
    reader    : InputStreamReader
    url       : URL
Algoritma
    ping to google
    if(reachable)
        add google result to listModel
    endif
endprosedur
```

Gambar 4.22. Prosedur google_search()

Prosedur *google_search()* pada Gambar 4.22 digunakan untuk melakukan pencarian file-file di mesin pencari Google. Proses pencarian dilakukan berdasarkan kata kunci yang sudah disimpan sebelumnya yang dikarenakan tidak ada satupun file berdasarkan kata kunci yang ditemukan pada prosedur *search()*. Program akan memetakan 10 pencarian teratas menggunakan *filter* “filetype:pdf” untuk selanjutnya

dapat dilakukan proses pengunduhan file pada prosedur *download_file()* yang ditunjukkan pada Gambar 4.24.

```
prosedur copy_file(input sourcePath: String, destinationPath: String)
{ copy file dari folder}

Deklarasi
    reader : InputStream
    writer : FileOutputStream
    url    : URL

Algoritma
    reader ← url.openStream
    writer ← destinationPath + sourcePath_name
    while(reader read buffer) do
            writer write buffer
    endwhile
        close reader
        close writer
endprosedur
```

Gambar 4.23. Prosedur copy_file()

Prosedur *copy_file()* yang ditunjukkan pada Gambar 4.23 digunakan untuk menyalin file-file PDF sesuai dengan hasil pencarian pada prosedur *search()*. Hasil penyalinan disimpan pada sebuah folder khusus.

```
prosedur download_file(input sourcePath: String,
destinationPath: String)
{ download file dari Internet }

Deklarasi
    reader: InputStream
    writer: FileOutputStream
    url   : URL

Algoritma
    reader ← url.openStream
    writer ← destinationPath + sourcePath_name
    while(reader read buffer) do
            writer write buffer
    endwhile
        close reader
        close writer
endprosedur
```

Gambar 4.24. Prosedur download_file()

Prosedur *download_file()* yang ditunjukkan pada Gambar 4.24 digunakan untuk mengunduh file-file PDF sesuai dengan hasil pencarian pada prosedur google_*search().* Hasil pengunduhan disimpan pada sebuah folder khusus.

4.2.4 Implementasi Antarmuka

Antarmuka aplikasi didasari pada Interaksi Manusia dan Komputer (IMK) yang baik dan sangat sederhana, sesuai kondisi sosial masyarakat pedesaan karena tidak semua orang di daerah pedesaan mengerti cara kerja

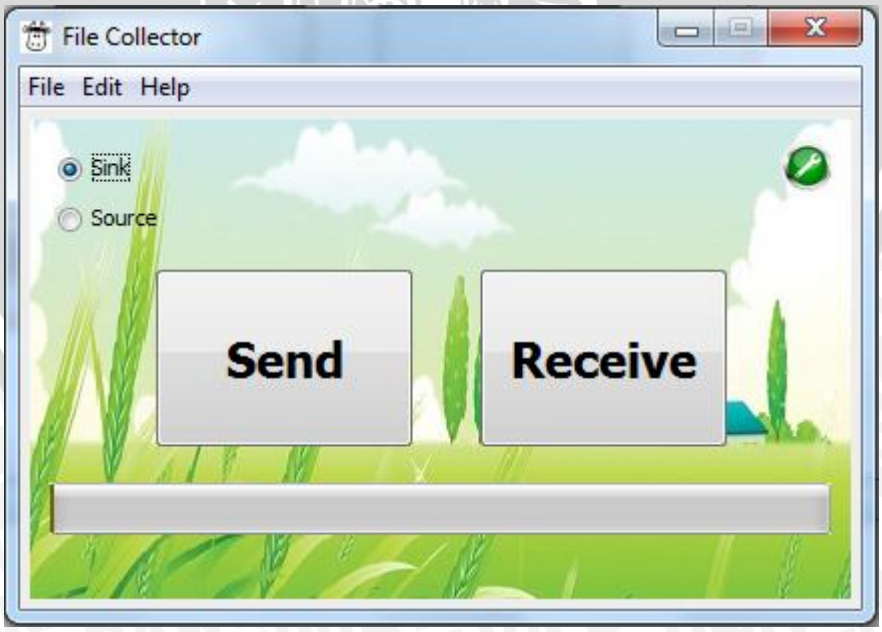
perangkat lunak. Dengan tampilan yang sederhana diharapkan penggunaan aplikasi menjadi mudah dimengerti.

- **File Browser**

Gambar 4.25. Antarmuka Utama File Browser

Gambar 4.25 merupakan antarmuka File Browser. Pada halaman ini terdapat kolom pencarian dan tombol pencarian yang digunakan oleh *user*. Tombol permintaan akan ditampilkan apabila *user* telah melakukan pencarian. Halaman ini dapat diakses melaui *http://localhost web server* itu sendiri atau mengakses alamat IP server dalam LAN.

- **File Collector**



Gambar 4.26. Antarmuka Utama File Collector

Gambar 4.26 merupakan antarmuka File Collector. Pada tampilan aplikasi, terdapat dua buah *radio button* yaitu tombol *Sink* dan tombol *Source*. Masing-masing tombol tersebut menunjukkan lokasi dimana kurir data sedang berada, yaitu tombol *Sink* ketika berada di Sink POA dan tombol *Source* ketika berada di Source POA. Tombol *Send* dan *Receive* menunjukkan aksi yang harus dilakukan oleh kurir data di masing-masing lokasi.

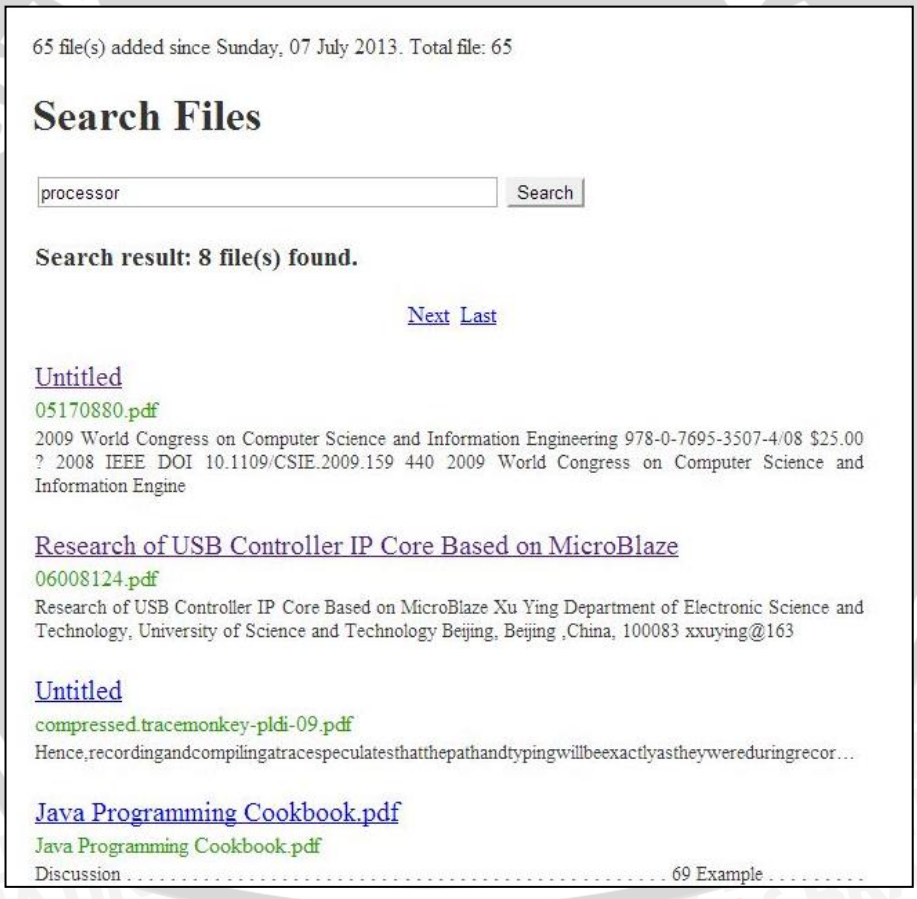
BAB V
PENGUJIAN DAN ANALISIS

5.1 Pengujian

5.1.1 Pengujian Kebutuhan Fungsional

Pada penelitian ini, *Black Box Testing* digunakan sebagai metode pengujian fungsional sistem untuk menjamin kerja aplikasi sesuai dengan kebutuhan serta meminimalisir kecacatan dalam sistem, sehingga tidak mempersulit pengguna saat pemakaian. Pengujian ini dilakukan dengan menginspeksi masing-masing komponen dan fungsionalitas aplikasi.

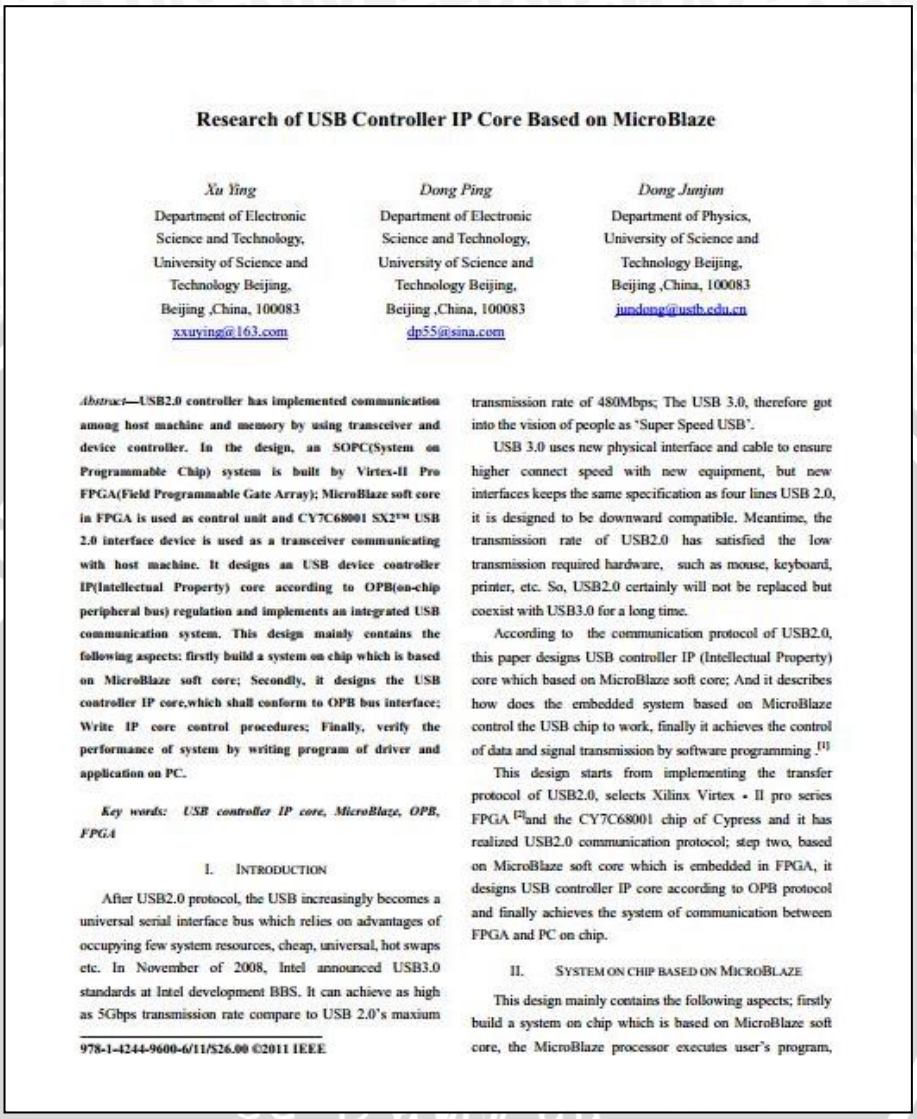
- **File Browser**



Gambar 5.1. Antarmuka: Hasil Pencarian File

Gambar 5.1 menunjukkan hasil pencarian dari kata kunci yang dimasukkan oleh *user* saat tombol *Search* ditekan. Hasil keluaran berupa judul file, nama file, dan isi file pada bagian awalnya saja. Banyaknya

hasil pencarian tergantung dari jumlah file yang mengandung kata kunci bersangkutan.



Research of USB Controller IP Core Based on MicroBlaze

Xu Ying
Department of Electronic Science and Technology, University of Science and Technology Beijing, Beijing ,China, 100083
xxuying@163.com

Dong Ping
Department of Electronic Science and Technology, University of Science and Technology Beijing, Beijing ,China, 100083
dp55@sina.com

Dong Junjun
Department of Physics, University of Science and Technology Beijing, Beijing ,China, 100083
jundong@ustb.edu.cn

***Abstract*—USB2.0 controller has implemented communication among host machine and memory by using transceiver and device controller. In the design, an SOPC(System on Programmable Chip) system is built by Virtex-II Pro FPGA(Field Programmable Gate Array); MicroBlaze soft core in FPGA is used as control unit and CY7C68001 SX2™ USB 2.0 interface device is used as a transceiver communicating with host machine. It designs an USB device controller IP(Intellectual Property) core according to OPB(on-chip peripheral bus) regulation and implements an integrated USB communication system. This design mainly contains the following aspects: firstly build a system on chip which is based on MicroBlaze soft core; Secondly, it designs the USB controller IP core,which shall conform to OPB bus interface; Write IP core control procedures; Finally, verify the performance of system by writing program of driver and application on PC.**

***Key words:** USB controller IP core, MicroBlaze, OPB, FPGA*

I. INTRODUCTION

After USB2.0 protocol, the USB increasingly becomes a universal serial interface bus which relies on advantages of occupying few system resources, cheap, universal, hot swaps etc. In November of 2008, Intel announced USB3.0 standards at Intel development BBS. It can achieve as high as 5Gbps transmission rate compare to USB 2.0's maxium transmission rate of 480Mbps; The USB 3.0, therefore got into the vision of people as 'Super Speed USB'.

USB 3.0 uses new physical interface and cable to ensure higher connect speed with new equipment, but new interfaces keeps the same specification as four lines USB 2.0, it is designed to be downward compatible. Meantime, the transmission rate of USB2.0 has satisfied the low transmission required hardware, such as mouse, keyboard, printer, etc. So, USB2.0 certainly will not be replaced but coexist with USB3.0 for a long time.

According to the communication protocol of USB2.0, this paper designs USB controller IP (Intellectual Property) core which based on MicroBlaze soft core; And it describes how does the embedded system based on MicroBlaze control the USB chip to work, finally it achieves the control of data and signal transmission by software programming.[1]

This design starts from implementing the transfer protocol of USB2.0, selects Xilinx Virtex - II pro series FPGA [2]and the CY7C68001 chip of Cypress and it has realized USB2.0 communication protocol; step two, based on MicroBlaze soft core which is embedded in FPGA, it designs USB controller IP core according to OPB protocol and finally achieves the system of communication between FPGA and PC on chip.

II. SYSTEM ON CHIP BASED ON MICROBLAZE

This design mainly contains the following aspects; firstly build a system on chip which is based on MicroBlaze soft core, the MicroBlaze processor executes user's program,

978-1-4244-9600-6/11/$26.00 ©2011 IEEE

Gambar 5.2. Antarmuka: Baca File

Gambar 5.2 menunjukkan tampilan salah satu file PDF yang telah dipilih dari beberapa hasil pencarian dan dibuka oleh PDF Reader.

Gambar 5.3. Antarmuka: Tambah Permintaan Berhasil

Gambar 5.3 menunjukkan bahwa permintaan *user* terhadap file sesuai kata kunci berhasil saat tombol permintaan ditekan. Aplikasi File Browser memberitahukan *user* bahwa file akan sampai dalam beberapa hari.

Gambar 5.4. Antarmuka: Permintaan Sudah Ada

Gambar 5.4 menunjukkan bahwa permintaan *user* mengalami kegagalan saat tombol permintaan ditekan. Aplikasi File Browser memberitahukan *user* bahwa kata kunci yang dimasukkan sudah pernah diminta oleh *user* itu sendiri atau *user* lain.

Gambar 5.5. Antarmuka: Permintaan Tidak Sesuai Kosa Kata

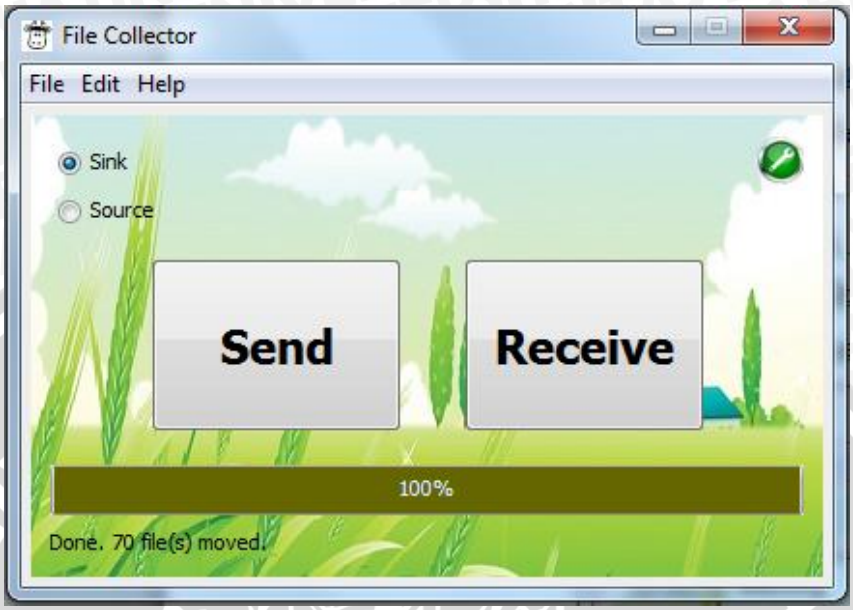
Gambar 5.5 menunjukkan bahwa permintaan *user* mengalami kegagalan saat tombol permintaan ditekan. Aplikasi File Browser memberitahukan *user* bahwa kata kunci yang dimasukkan tidak sesuai dengan kosa kata Bahasa Inggris maupun Bahasa Indonesia.

No.	Nama Pengujian	Kesimpulan	
		Valid	Tidak Valid
1.	Search Files	√	
2.	Add Request	√	
3.	Pagination	√	
4.	Read File	√	
5.	Request Input	√	

Tabel 5.1. Hasil Pengujian Validasi File Browser

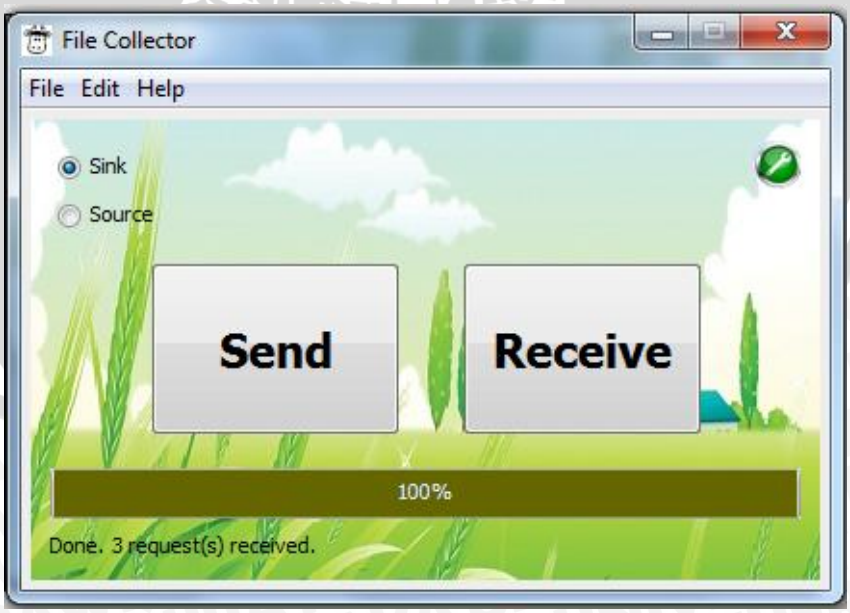
Tabel 5.1 merupakan hasil pengujian validasi File Browser. Hasil dari pengujian validasi menunjukkan bahwa seluruh kebutuhan komponen antarmuka dan fungsionalitas File Browser adalah valid.

- **File Collector**
  - **Sink POA**



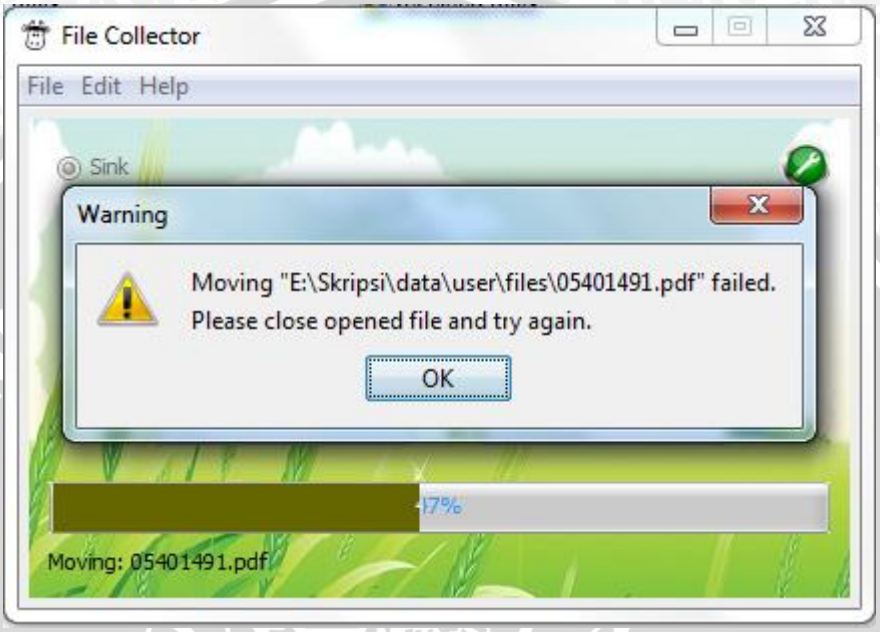
Gambar 5.6. Antarmuka: Pemindahan File

Gambar 5.6 menunjukkan bahwa aplikasi File Collector berhasil memindahkan file-file hasil pencariannya di Source POA ke sebuah folder di File Browser di Sink POA saat tombol *Send* ditekan.



Gambar 5.7. Antarmuka: Penerimaan Permintaan

Gambar 5.7 menunjukkan bahwa aplikasi File Collector berhasil menerima file-file permintaan oleh *user* dari File Browser di Sink POA saat tombol *Receive* ditekan. Semua permintaan yang didapatkan disimpan pada sebuah folder khusus.

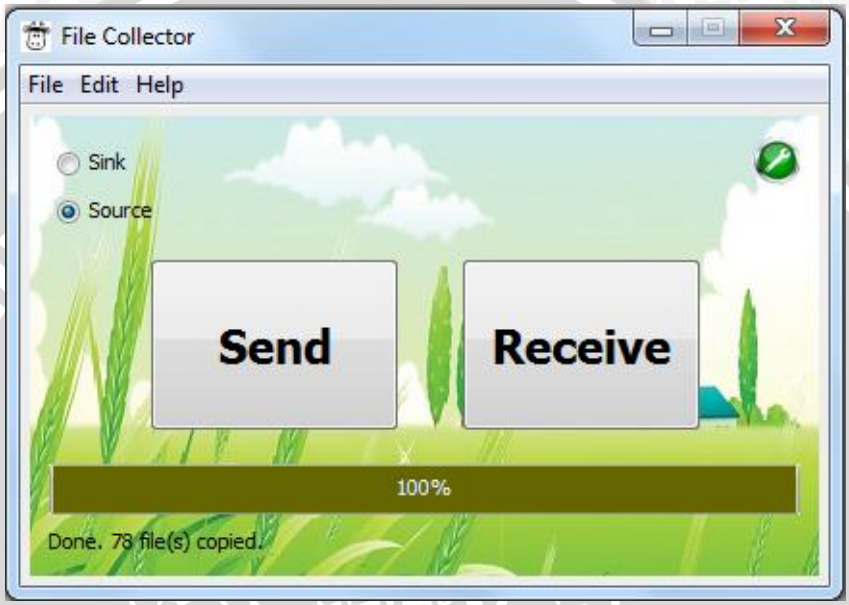


Gambar 5.8. Antarmuka: File Dibuka Aplikasi Lain

Gambar 5.8 menunjukkan hasil pengujian fungsional dalam hubungannya dengan penguncian sumber daya oleh aplikasi lain. Pengujian ini dilakukan untuk mengetahui tingkah laku aplikasi saat ada file yang sedang digunakan oleh aplikasi lain. Operasi file yang diujikan adalah operasi *move* (tombol *Send*) karena pada operasi ini, penguncian sumber daya oleh aplikasi lain akan menghalangi proses pemindahan file. Tiap aplikasi memiliki tingkah laku yang berbeda dalam proses penguncian sumber daya, sehingga ada proses pemindahan file yang sukses dan tidak sukses. Gambar 5.8 menunjukkan bahwa apabila proses pemindahan file tidak sukses, maka aplikasi memberikan solusi kepada kurir data. Pada aplikasi File Collector, dilakukan pengujian dengan membuka sebuah file sebelum file tersebut dipindahkan. Hasil dari pengujian ini

menunjukkan apabila ada file yang dibuka, maka file tersebut tidak dapat dipindahkan, tetapi bila aplikasi lain tidak lagi menggunakannya, maka file dapat dipindahkan dengan kembali menekan tombol *Send*.

- **Source POA**



Gambar 5.9. Antarmuka: Penyalinan File

Gambar 5.9 menunjukkan bahwa aplikasi File Collector berhasil menyalin file-file hasil pencariannya ke sebuah folder khusus pada USB Flash Drive.

Gambar 5.10. Antarmuka: Free Space Memory Tidak Cukup

Gambar 5.10 menunjukkan hasil pengujian fungsional dalam hubungannya dengan ketersediaan *free space memory*. Pengujian ini dilakukan untuk menguji keandalan sistem saat kondisi USB Flash Drive yang digunakan oleh kurir data tidak memiliki cukup ruang kosong saat penyalinan/pengunduhan file-file. Pengujian ini dilakukan dengan cara mengisi USB Flash Drive dengan data hingga hampir penuh. Saat dilakukan penyalinan/pengunduhan, maka dapat dilihat tindakan yang diambil oleh aplikasi File Collector. File Collector akan menampilkan pesan peringatan yang meminta kurir data untuk menghapus beberapa data agar USB Flash Drive tersebut memiliki cukup ruang kosong. Pesan peringatan akan tetap muncul selama proses pengulangan sampai *free space memory* mencukupi.

No.	Nama Pengujian	Kesimpulan	
		Valid	Tidak Valid
1.	Send Files	√	
2.	Receive XMLs	√	
3.	Receive Files	√	
4.	Change Search Target	√	
5.	Enable/Disable Component	√	

Tabel 5.2. Hasil Pengujian Validasi Komponen File Collector

Tabel 5.2 merupakan hasil pengujian validasi File Collector. Hasil dari pengujian validasi menunjukkan bahwa seluruh kebutuhan komponen antarmuka dan fungsionalitas File Collector adalah valid.

5.1.2 Pengujian Kebutuhan Non-Fungsional

Pengujian non-fungsional pada aplikasi ini meliputi pengujian terhadap hal-hal yang mungkin terjadi saat sistem dioperasikan diantaranya:

- Pengujian Ketersediaan Listrik.

  Pengujian ini dilakukan untuk menguji keandalan sistem saat kondisi listrik padam secara tiba-tiba. Saat dilakukan pemindahan file oleh File Collector, aplikasi ditutup secara paksa (sama ketika listrik mati), kemudian aplikasi dijalankan lagi dan mengaktifkan kembali proses pemindahan file yang tertunda sebelumnya.

  Hasil dari pengujian ini menunjukkan bahwa file-file yang telah dipindahkan akan tetap berada di dalam folder tujuan. Pada saat proses pemindahan file dilakukan kembali, maka proses dimulai dari file terakhir saat pemindahan file berlangsung ketika listrik padam.

- Pengujian Keandalan Sistem

  Pengujian keandalan sistem adalah pengujian yang dilakukan untuk mengetahui keandalan sistem saat ada kejadian tak terduga. Dalam kasus ini, kejadian tersebut adalah saat USB Flash Drive yang dibawa oleh kurir data hilang atau rusak dalam perjalanan ke Source POA. Data tersebut berupa file-file permintaan oleh *user* di Sink POA. Data yang hilang (yang dikarenakan USB Flash Drive yang hilang) seharusnya dapat diambil kembali oleh kurir data saat kembali ke Sink POA.

  Ketika permintaan-permintaan dalam bentuk file-file XML diambil dari Sink POA, maka File Collector tidak boleh menghapus file-file tersebut. File-file permintaan hanya boleh dihapus hanya bila file-file PDF telah didapatkan dari Source POA. Pengujian dilakukan dengan cara sengaja mengosongkan folder permintaan dari *user* di Sink POA dan memintanya kembali dengan tombol *Receive*. Hasil dari pengujian ini menunjukkan bahwa seluruh file permintaan disalin kembali ke sebuah folder khusus di USB Flash Drive.

5.2 Analisis

Analisis terhadap sistem berbagi informasi digital pedesaan menunjukkan kesesuaian hasil pengujian dengan kebutuhan sistem. Analisis hasil pengujian terdiri dari analisis kebutuhan fungsional dan non-fungsional.

5.2.1 Analisis Kebutuhan Fungsional

Bagian ini menunjukkan beberapa analisis hasil pengujian kebutuhan fungsional terhadap seluruh komponen dan fungsi masing-masing aplikasi yang ada di sistem berbagi informasi digital pedesaan.

1. Pengujian validasi terhadap komponen antarmuka pada desain aplikasi File Browser dan File Collector menunjukkan seluruh komponen antarmuka sudah lengkap dan dapat berfungsi dengan baik sesuai kebutuhan sistem.
2. Secara keseluruhan, baik aplikasi File Browser maupun File Collector memiliki antarmuka pengguna yang baik dan mudah dimengerti penggunaannya.

3. Proses permintaan file oleh *user* di File Browser akan mengalami kegagalan apabila kata kunci yang dimasukkan oleh *user* tidak valid yang terjadi karena dua kondisi diantaranya:
   - Kata kunci yang dimasukkan tidak terdaftar di dalam kosa kata baik dalam Bahasa Inggris maupun Bahasa Indonesia. pada basis data.
   - Kata kunci yang dimasukkan berisi imbuhan (dalam Bahasa Indonesia). Hal ini dikarenakan daftar kosa kata hanya mengandung kata dasar saja. Untuk kata kunci dalam Bahasa Inggris, hal ini tidak berlaku dikarenakan daftar kosa kata dalam Bahasa Inggris sudah lengkap.

5.2.2 Analisis Kebutuhan Non-Fungsional

Hasil pengujian kebutuhan non-fungsional menunjukkan keandalan sistem berbagi informasi digital pada saat implementasi diantaranya:.

1. Sistem mampu menangani kondisi ketika listrik mati saat implementasi sistem. Proses pemindahan file-file akan dimulai kembali pada kondisi terakhir pemindahan ketika listrik mati secara mendadak. Aplikasi tidak perlu mengulang proses dari awal.
2. Sistem dapat menangani kondisi dimana USB Flash Drive yang dibawa oleh kurir data hilang atau. Ketika USB Flash Drive yang dibawa oleh kurir data hilang atau rusak, maka kurir tersebut dapat kembali mengambil permintaan-permintaan *user* di Sink POA, karena file-file tersebut masih disimpan di server lokal. File-file permintaan hanya akan dihapus apabila file-file PDF yang diinginkan sudah didapatkan (sesuai permintaan di file-file permintaan). Selain itu, seorang kurir data juga harus memiliki akses transportasi dari Sink POA ke Source POA dan sebaliknya.

Skenario penggunaan sistem berbagi informasi digital di daerah terpencil menunjukkan bahwa informasi (dalam hal ini adalah data digital dalam bentuk file) yang dikirimkan dari sumber data (Source POA) ke

tujuan (Sink POA) merupakan data-data utuh. Mekanisme pengiriman data pada sistem ini berbeda dengan mekanisme pengiriman data pada VDTN, dimana data dipecah menjadi beberapa bagian pada sumber data yang kemudian dikumpulkan lagi di tujuan. Kurir data bertindak sebagai pengantar informasi digital dengan menggunakan media penyimpanan data sebagai alat pembawa informasi ini. Hal inilah yang mengatasi masalah *store and forward* pada VDTN yang tidak efektif apabila diterapkan pada daerah yang sangat terpencil.

BAB VI
PENUTUP

Berdasarkan hasil perancangan, implementasi, dan pengujian yang dilakukan pada sistem berbagi informasi digital, maka dapat diambil kesimpulan secara keseluruhan dan saran-saran untuk pengembangan sistem selanjutnya.

6.1 Kesimpulan

1. Dalam penerapannya, sistem yang dibuat dapat bekerja secara efektif apabila ada dua hal yang terpenuhi. Pertama, seorang *user* memiliki akses ke *local web server* yang ada di desa, misalkan di balai desa. Kedua, kurir data harus memiliki akses ke server di sumber data dengan alat transportasi apapun.
2. Hasil evaluasi terhadap prototipe sistem menunjukkan bahwa sistem dapat bekerja dengan baik pada kondisi ketersediaan listrik yang tidak menentu, memiliki antarmuka yang memudahkan pengguna, dan memiliki keandalan terhadap kondisi dimana media penyimpanan data mengalami kerusakan atau hilang.
3. Prototipe sistem dalam penelitian ini dapat mengatasi *digital divide* antara pedesaan dan perkotaan dan menjadi alternatif pada daerah dengan ketiadaan koneksi *end-to-end*. Dalam implementasinya, sistem ini dapat mengatasi kekurangan mekanisme *store and forward* pada VDTN yang kurang efektif apabila diterapkan pada daerah yang sangat terpencil.

6.2 Saran

1. Sistem berbagi informasi digital diharapkan nantinya tidak hanya menyediakan informasi yang ada pada file PDF, tetapi juga dapat mendukung beberapa tipe file lainnya yang berkaitan dengan pengembangan Sumber Daya Manusia (SDM) di daerah pedesaan. File-file gambar dan artikel-artikel dalam bentuk file HTML misalnya, juga merupakan informasi yang penting untuk disebarluaskan di daerah pedesaan.
2. Pencarian file-file di Internet yang dilakukan oleh File Collector sebaiknya juga dapat mengetahui file mana saja yang sudah pernah diunduh dan mencari file lain yang sejenis, sehingga tidak menimbulkan pemborosan waktu pencarian di sumber data.
3. Sistem berbagi informasi digital diharapkan nantinya dapat dioperasikan di beberapa *platform*. Kebutuhan perangkat keras dan perangkat lunak dapat disesuaikan dengan kondisi pedesaan masing-masing.

DAFTAR PUSTAKA

[DWS-13] Darwin, Waizly. 2012, Survey MarkPlus Insight: Naik 6 Juta, Pengguna Internet Indonesia kini 61 Juta Orang. http://the-marketeers.com/archives/indonesia-Internet-users-waizly-2012.html [diakses 25 Juni 2013].

[DMV-13] Dahlan, BAB 7 Pengenalan Arsitektur MVC – Dahlan Unimal, http://dahlan.unimal.ac.id/files/modul-web/JENI-Web%20Programming-Bab%207-MVC%20Intro.pdf [diakses 20 Mei 2013]

[FKD-03] Fall, Kevin. 2003, "A Delay-Tolerant Network Architecture for Challenged Internets", Intel Research, Berkeley.

[JVD-13] Joel J. P. C. Rodrigues, João A. Dias, João N. Isento, Bruno M. Silva, Vasco N. G. J. Soares, Naércio Magaia, Paulo Rogério Pereira, Augusto Casaca, Cristina Cervelló-Pastor, Javier Gallego, "The Vehicular Delay-Tolerant Networks (VDTN) Euro-NF Joint Research Project".

[IMD-13] Kawasan Ekonomi Khusus & Internet Masuk Desa. http://fkppit.blogspot.com/2009/06/kawasan-ekonomi-khusus-Internet-masuk.html [diakses 25 Juni 2013].

[API-13] Ari, Program Internet Masuk Desa Belum Maksimal. http://musirawasekspres.blogspot.com/2013/05/program-Internet-masuk-desa-belum.html [diakses 25 Juni 2013].

[JMX-03] Junaedi, Moh. 2003, "Pengantar XML", IlmuKomputer.Com.

[XML-05] Dykes, Lucinda dan Tittel, Ed. 2005, XML For Dummies®, 4th Edition. Wiley Publishing, Inc., Indianapolis, Indiana

[ASM-13] Solichin, Achmad. Pemrograman Web dengan PHP dan MySQL. Universitas Budi Luhur, Jakarta.

[ASI-06] Adobe Systems Incorporated. 2006, PDF Reference, sixth edition: Adobe Portable Document Format version 1.7.

[VNI-13] Vitriana, Novita. "Internet dan Perpustakaan", Pustakawan Pelaksana pada UPT Perpustakaan Universitas Sriwijaya.

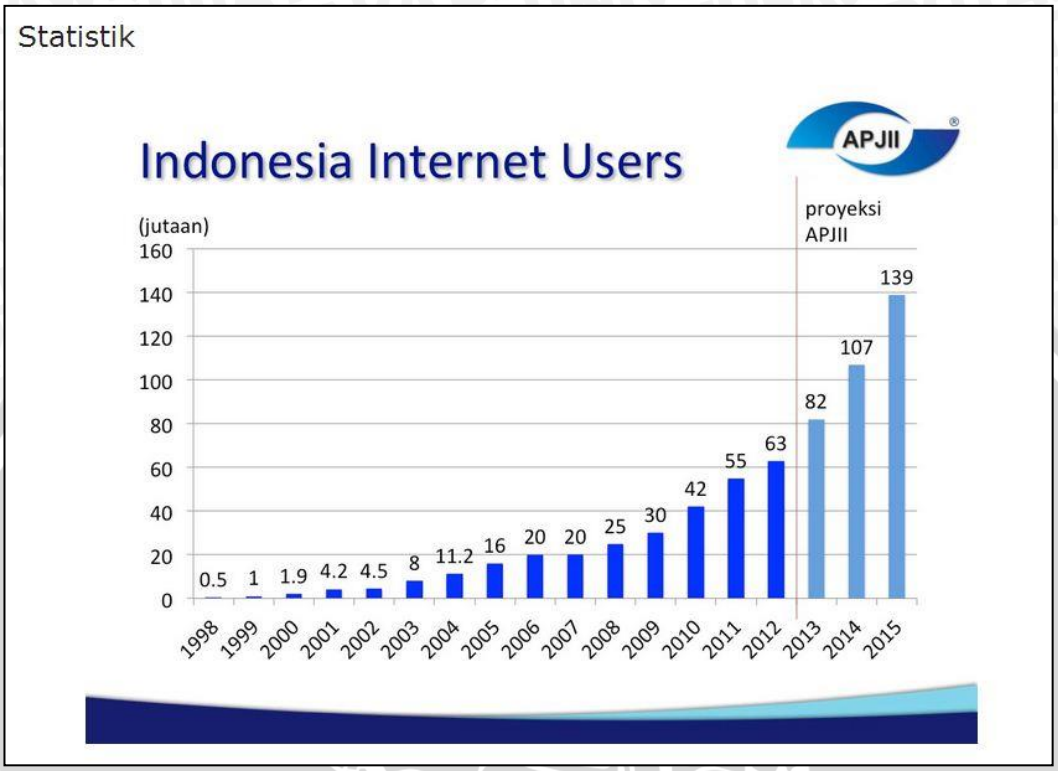
[AWS-07] Azmi, Fauzan. "Web Server", http://blog.azmifauzan.net/wp-content/uploads/2007/10/web-server.pdf [diakses 26 Juni 2013].

[APJ-12] Asosiasi Penyelenggara Jasa Internet Indonesia (APJII). "Profil Pengguna Internet Indonesia 2012".

[PRD-13] Palapa Ring. http://www.detiknas.org/index.php/flagship/c/14/ [diakses 25 Juni 2013].

[BKC-05] Barrett, Diane and King, Todd. 2005, Computer Networking Illuminated. Jones and Bartlett Publishers, Inc., Sudbury, MA.

LAMPIRAN 1



Gambar 1.1. Statistik Pengguna Internet di Indonesia

Sumber: [APJ-12]

LAMPIRAN 2

Telkom Fixed Line/Rumah Tangga	Total	Status	
		Rural	Kota
Total Kelurahan	69.957	57.667 (82%)	12.290 (18%)
Kelurahan dengan akses kabel Telkom	34,68 %	23,03 %	89,32 %
Kelurahan yang memiliki warnet	3,04 %	0,62 %	14,39 %
Kelurahan dengan sinyal telepon seluler kuat	41,03 %	30,95 %	88,32 %
Kelurahan dengan sinyal telepon seluler lemah	29,56 %	33,88 %	9,30 %
Kelurahan tidak ada sinyal telepon seluler	29,41 %	35,17 %	2,38 %
Total	**100,00 %**	**100,00**	**100,00**

Gambar 2.1. Data Akses Internet di Kelurahan di Indonesia

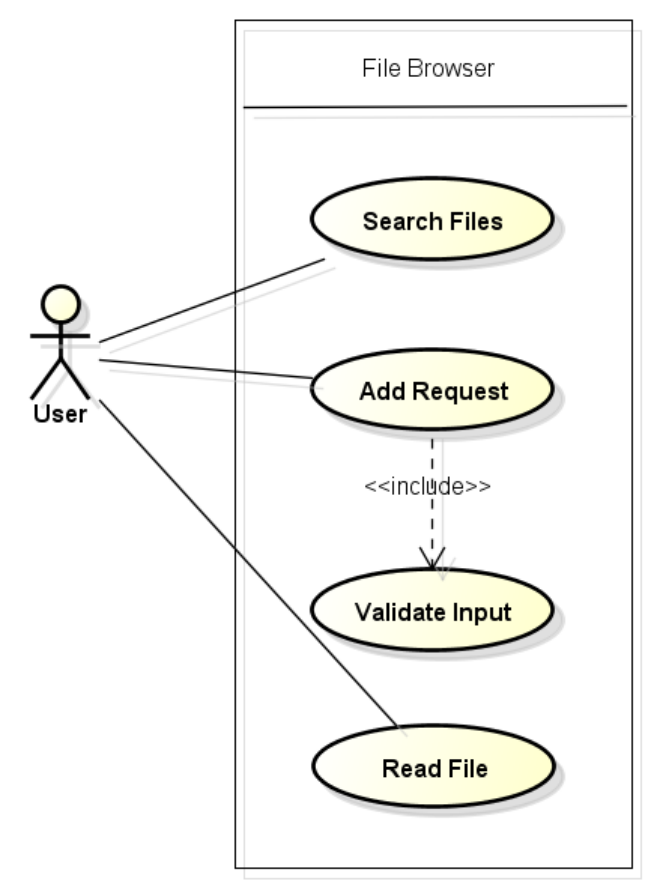
Sumber: [APJ-12]

LAMPIRAN 3
Detail Desain Aplikasi

1 Use Case Diagram

Use Case Diagram menunjukkan apa saja yang dapat dilakukan oleh *actor* (dalam hal ini, yaitu *user* dan kurir data) pada masing-masing aplikasi yang digunakannya secara umum.

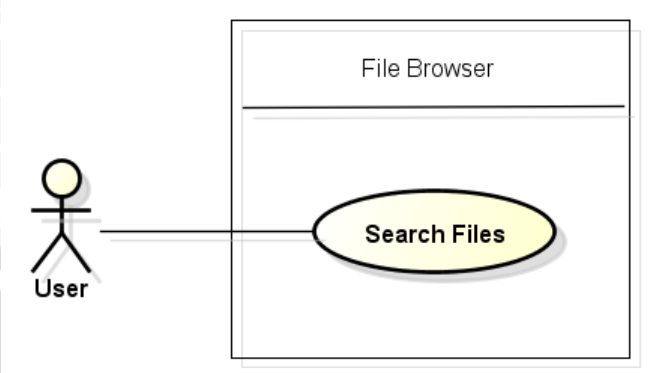
- ***User***



Gambar 3.1. Use Case: *User*

Gambar 3.1 menunjukkan bahwa *user* memiliki tiga aksi, yaitu melakukan pencarian file, menambah permintaan, dan membaca file.

- **Diagram: *Search Files***



Gambar 3.2. Use Case: Search Files

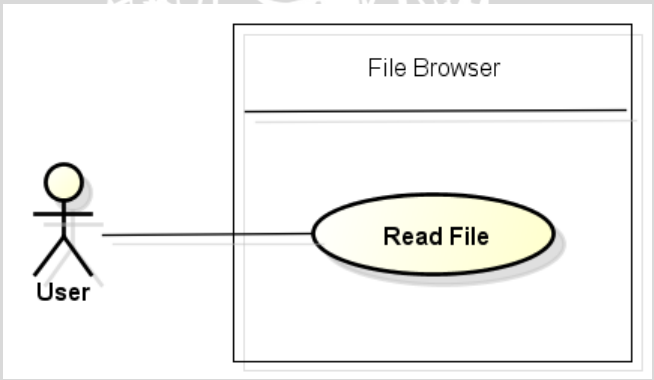
Deskripsi

- Use case ini menunjukkan bahwa *user* melakukan pencarian terhadap file.

Deskripsi langkah-langkah

- *User* memasukkan kata kunci sesuai deskripsi file yang diinginkan-nya.

- **Diagram: *Read File***



Gambar 3.3. Use Case: Read File

Deskripsi

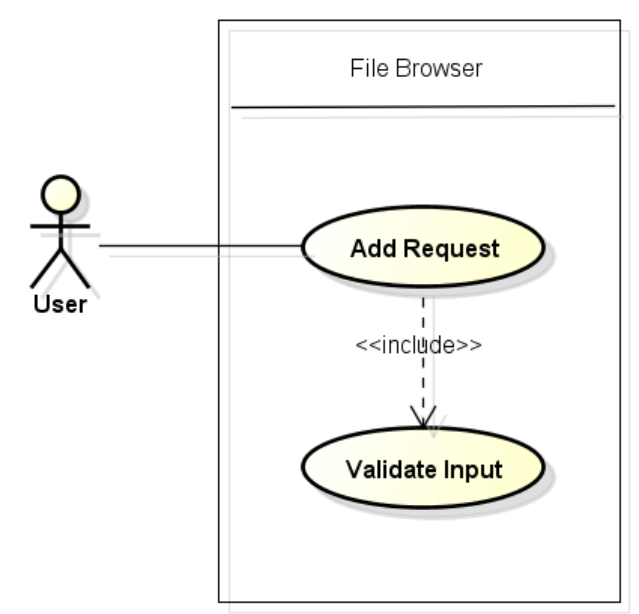
- Use case ini menunjukkan bahwa *user* membaca file.

Deskripsi langkah-langkah

- *User* melakukan pencarian file sesuai keinginannya.

- Memilih salah satu file hasil keluaran dari sistem yang akan menampilkan isi file.

- **Diagram: *Add Request***



Gambar 3.4. Use Case: Add Request

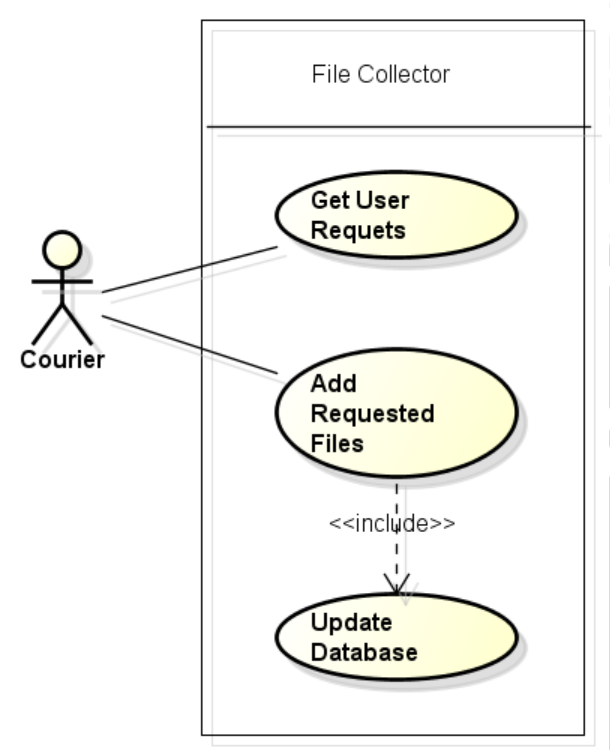
Deskripsi

- Use case ini menunjukkan bahwa *user* melakukan permintaan file kepada sistem.

Deskripsi langkah-langkah

- *User* menambah permintaan kepada sistem karena file yang dicari tidak ditemukan atau hasil tidak sesuai keinginan.

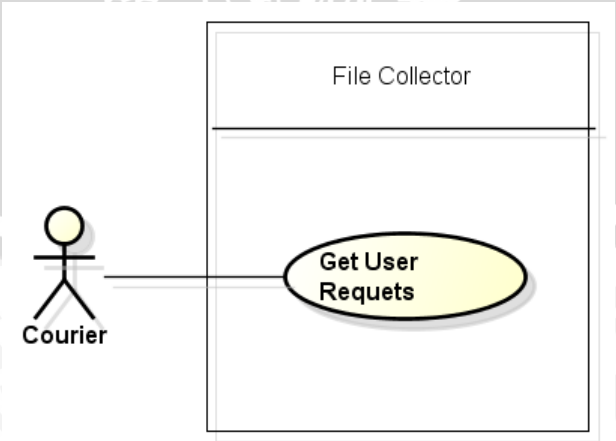
➢ **Kurir Data di Sink POA**



Gambar 3.5. Use Case: Kurir Data di Sink POA

Gambar 3.5 menunjukkan bahwa seorang kurir data di Sink POA memiliki tiga aksi, yaitu menerima permintaan *user*, menambah file-file permintaan, dan meng-*update* basis data.

- **Diagram: *Get User Request***



Gambar 3.6. Use Case: Get User Requests

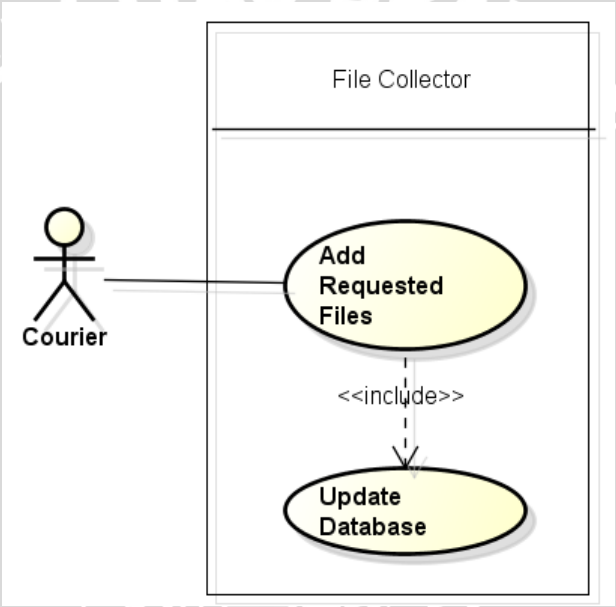
Deskripsi

- Use case ini menunjukkan bahwa kurir data menerima seluruh permintaan dari *user* pada Sink POA.

Deskripsi langkah-langkah

- Kurir data melakukan *attach* USB Drive.
- Aplikasi dijalankan untuk menerima permintaan *user*.

- **Diagram: *Add Requested Files***



Gambar 3.7. Use Case: Add Requested Files

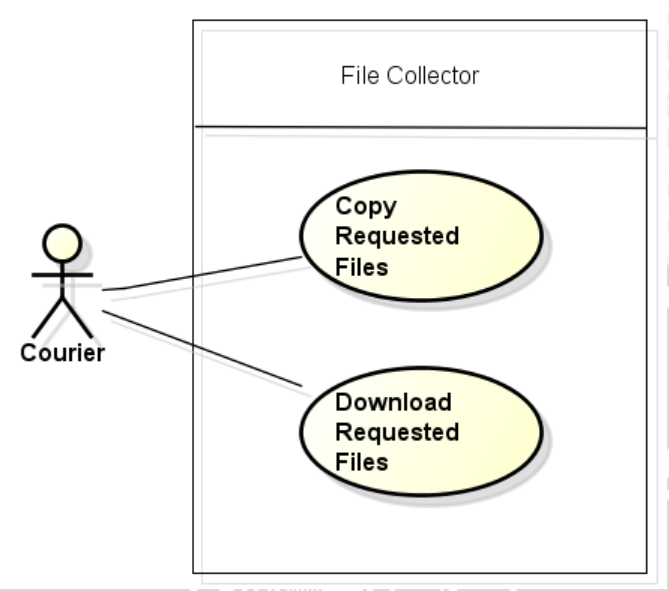
Deskripsi

- Use case ini menunjukkan bahwa kurir data memindahkan file hasil pencarian di sumber data dan menempatkannya di File Browser.

Deskripsi langkah-langkah

- Kurir data melakukan *attach* USB Drive.
- Aplikasi untuk memindahkan file ke direktori khusus File Browser. Aplikasi selanjutnya akan memperbaharui basis data secara otomatis.

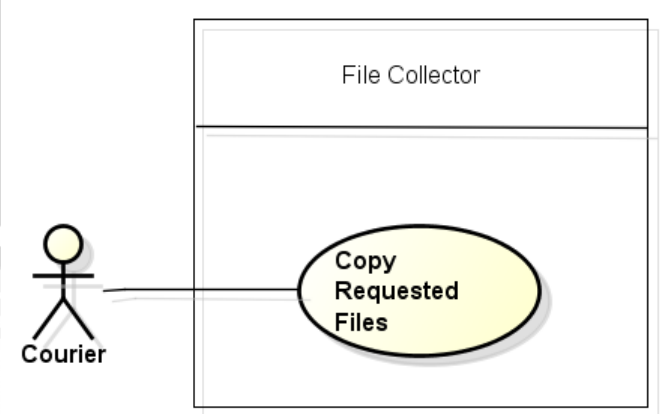
➢ **Kurir Data di Source POA**



Gambar 3.8. Use Case: Kurir Data di Source POA

Gambar 3.8 menunjukkan bahwa seorang kurir data di Source POA memiliki dua aksi, yaitu melakukan penyalinan dan pengunduhan file sesuai permintaan.

- **Diagram: *Copy Requested Files***



Gambar 3.9. Use Case: Copy Requested Files

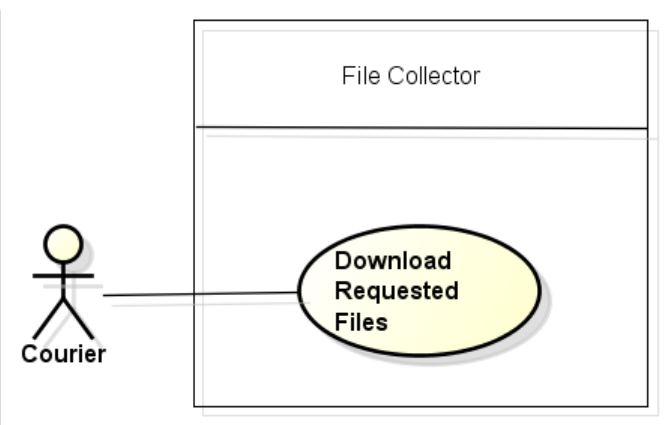
Deskripsi

- Use case ini menunjukkan bahwa kurir data memindahkan file hasil pencarian di sumber data dan menempatkannya di File Browser.

Deskripsi langkah-langkah

- Kurir data melakukan *attach* USB Drive.
  - Aplikasi dijalankan dan memilih menerima file-file sesuai permintaan yang tersimpan. File-file tersebut akan disalin ke direktori khusus.

- **Diagram: *Download Requested Files***



Gambar 3.10. Use Case: Download Requested Files

Deskripsi

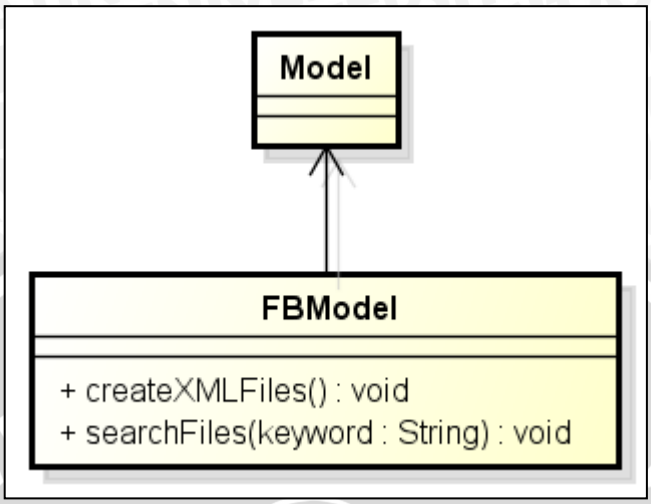
- Use case ini menunjukkan bahwa kurir data memindahkan file hasil pencarian di sumber data dan menempatkannya di File Browser.

Deskripsi langkah-langkah

- Kurir data melakukan *attach* USB Drive.
- Aplikasi dijalankan dan selanjutnya kurir data memilih untuk menerima file-file sesuai permintaan yang tersimpan. File-file tersebut akan diunduh ke direktori khusus bila ada koneksi Internet.

2 Class Diagram

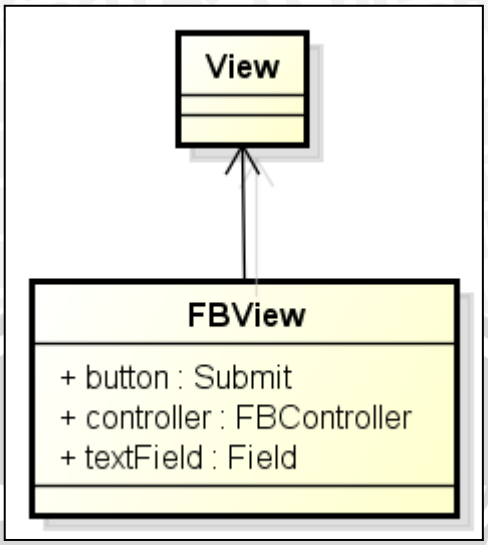
- **File Browser**



Gambar 3.11. Class Diagram: Model File Browser

Nama Kelas	**FBModel**	
Tipe	Model	
Deskripsi	Kelas ini digunakan sebagai representasi kelakuan program mengenai apa saja yang dapat dilakukan oleh aplikasi File Browser.	
Atribut	-	
Operasi	Nama	**createXMLFiles()**
	Argumen	-
	Kembalian	void
	Pre-condition	Tidak ada file
	Post-condition	File dalam format XML terbentuk
	Exception	-
	Aliran kejadian	- *User* mengisi kolom permintaan.
	Nama	**searchFiles()**
	Argumen	- keyword: String(100)
	Kembalian	void
	Pre-condition	Tidak ada
	Post-condition	Folder ditelusuri
	Exception	-
	Aliran kejadian	- *User* mengisi kolom pencarian.

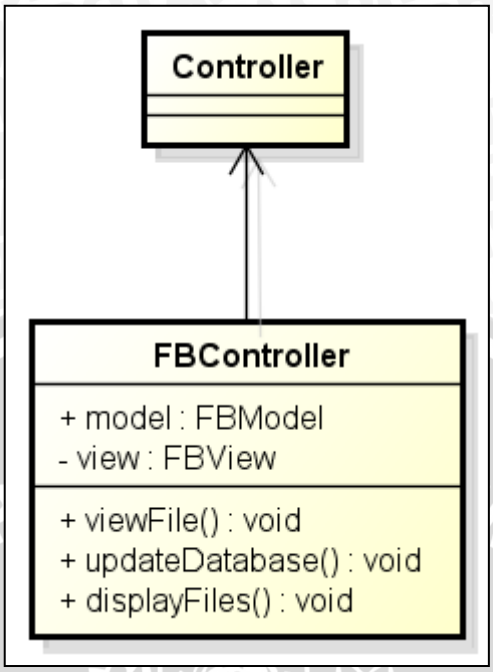
Tabel 3.1. Informasi Kelas FBModel



Gambar 3.12. Class Diagram: View File Browser

Nama Kelas	**FBView**
Tipe	View
Deskripsi	Kelas ini merepresentasikan tampilan dari aplikasi File Browser.
Atribut	- button: Submit - controller: FBController - textField: Field
Operasi	-

Tabel 3.2. Informasi Kelas FBView



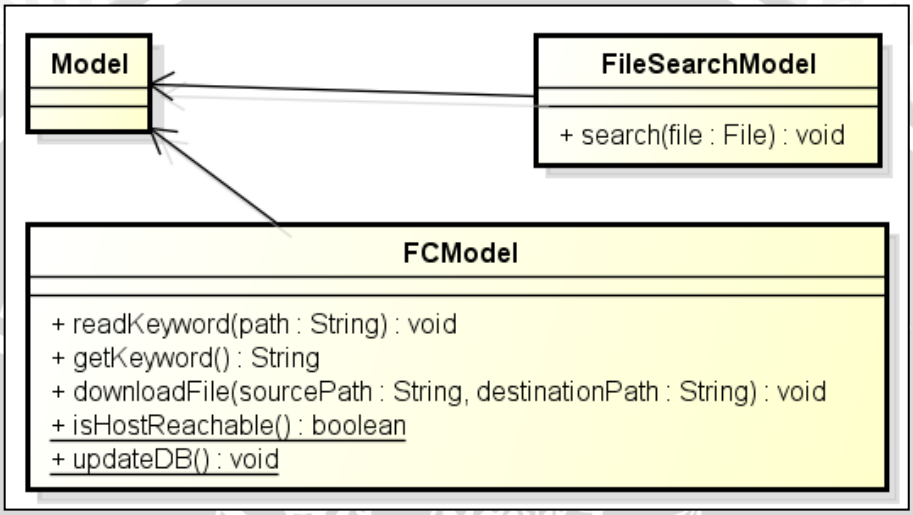
Gambar 3.13. Class Diagram: Controller File Browser

Nama Kelas	**FBController**	
Tipe	Controller	
Deskripsi	Kelas ini mengontrol interaksi yang terjadi antara kelas FBView dan kelas FBModel.	
Atribut	- model: FBModel - view: FBView	
Operasi	Nama	**viewFile()**
	Argumen	-
	Kembalian	void
	Pre-condition	Nama file ditampilkan dalam daftar
	Post-condition	Salah satu file ditampilkan untuk dibaca
	Exception	-
	Aliran kejadian	*User* memilih salah satu file dalam daftar
	Nama	**updateDatabase()**
	Argumen	-
	Kembalian	void
	Pre-condition	-
	Post-condition	Basis data ter-*update*
	Exception	-
	Aliran kejadian	- Kurir data memindahkan file ke server lokal
		- *Update* basis data terjadi secara otomatis
	Nama	**displayFiles()**
	Argumen	-
	Kembalian	void

	Pre-condition	Daftar pencarian kosong
	Post-condition	Ditampilkan hasil pencarian
	Exception	-
	Aliran kejadian	- *User* mengisi kolom pencarian

Tabel 3.3. Informasi Kelas FBController

- **File Collector**

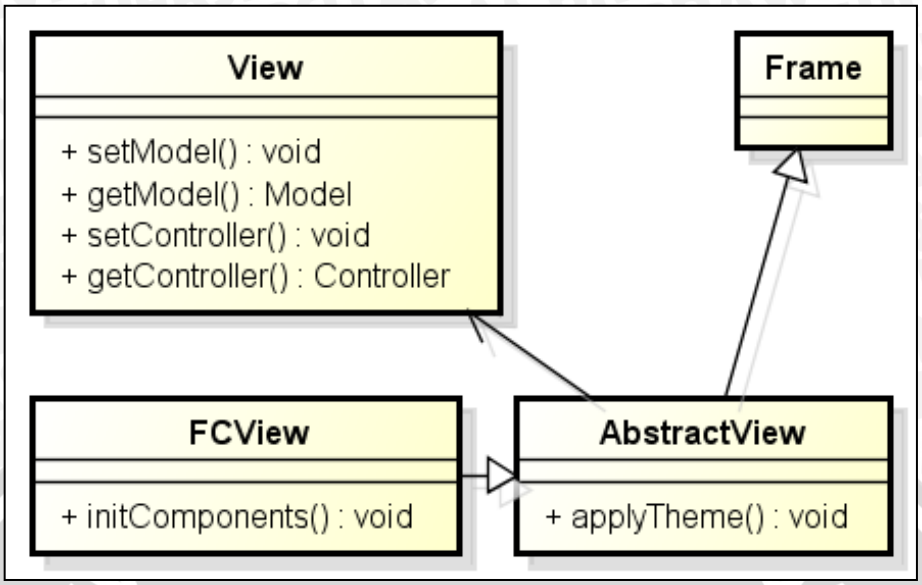


Gambar 3.14. Class Diagram: Model File Collector

Nama Kelas	**FCModel**	
Tipe	Model	
Deskripsi	Kelas ini digunakan sebagai representasi kelakuan program mengenai apa saja yang dapat dilakukan oleh aplikasi File Collector.	
Atribut	-	
Operasi	Nama	**readKeyword()**
	Argumen	path: String
	Kembalian	void
	Pre-condition	-
	Post-condition	Kata kunci ter-ekstrak dari file yang memuat-nya.
	Exception	- NullPointerException - ParserConfigurationException - SAXException - IOException
	Aliran kejadian	- Kurir data memilih pencarian file sesuai kata kunci hasil ekstraksi.

	Nama	**getKeyword()**
	Argumen	-
	Kembalian	String
	Pre-condition	-
	Post-condition	Kata kunci hasil ekstraksi didapatkan
	Exception	-
	Aliran kejadian	- Terjadi setelah proses pembacaan kata kunci
	Nama	**downloadFile()**
	Argumen	- sourcePath: String(100) - destinationPath: String()
	Kembalian	void
	Pre-condition	-
	Post-condition	Salinan file didapatkan
	Exception	- MalformedURLException - IOException
	Aliran kejadian	- Daftar file sesuai kata kunci ditemukan. - File-file otomatis terunduh satu-persatu.
	Nama	**isHostReachable()**
	Argumen	-
	Kembalian	boolean
	Pre-condition	-
	Post-condition	Ada tidaknya koneksi Internet diketahui.
	Exception	- IOException - InterruptedException
	Aliran kejadian	Kurir data memilih unduh file.
	Nama	**updateDB()**
	Argumen	-
	Kembalian	void
	Pre-condition	-
	Post-condition	Basis data ter-*update*
	Exception	- IOException - InterruptedException
	Aliran kejadian	- Kurir data memindahkan file ke server lokal.
		- *Update* basis data terjadi secara otomatis.

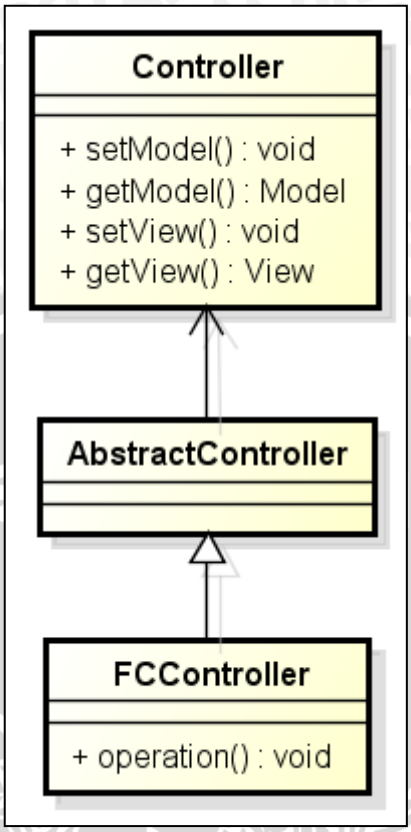
Tabel 3.4. Informasi Kelas FCModel



Gambar 3.15. Class Diagram: View File Collector

Nama Kelas	**FCView**	
Tipe	View	
Deskripsi	Kelas ini merepresentasikan tampilan dari aplikasi File Collector.	
Atribut	-	
Operasi	Nama	**initComponents()**
	Argumen	-
	Kembalian	void
	Pre-condition	Tampilan masih kosong
	Post-condition	Tampilan sudah terisi.
	Exception	-
	Aliran kejadian	Tampilan di-*load* otomatis ketika aplikasi dijalankan.

Tabel 3.5. Informasi Kelas FCView



Gambar 3.16. Class Diagram: Controller File Collector

Nama Kelas	**FCController**	
Tipe	Controller	
Deskripsi	Kelas ini mengontrol interaksi yang terjadi antara kelas FCView dan kelas FCModel.	
Atribut	-	
Operasi	Nama	**operation()**
	Argumen	-
	Kembalian	void
	Pre-condition	-
	Post-condition	Salah satu kejadian dalam fungsi terjadi
	Exception	-
	Aliran kejadian	- Kurir data memilih salah satu komponen aplikasi - Terjadi aksi sesuai pilihan.

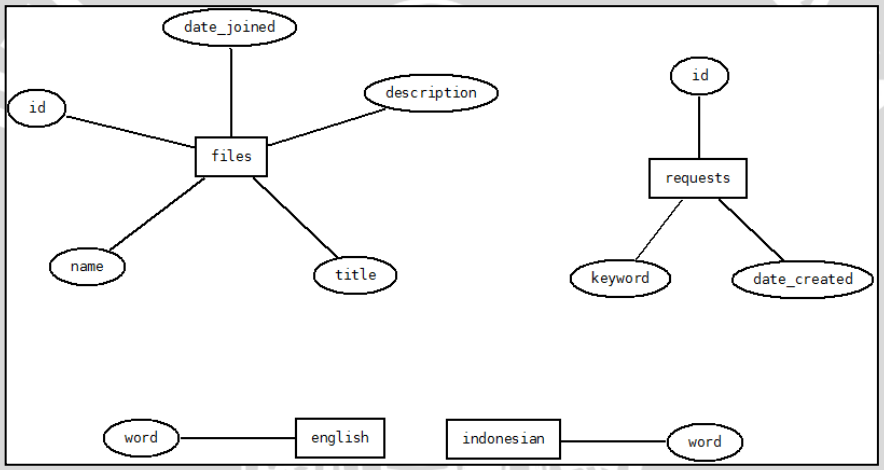
Tabel 3.6. Informasi Kelas FCController

3 Perancangan Basis Data

Perancangan basis data hanya dilakukan pada aplikasi File Browser. Perancangan basis data meliputi:

3.1 Entity Diagram

Entitas yang ada terdiri dari entitas *files*, *requests*, *english*, dan *indonesian*. Entitas *files* berisi informasi berupa *metadata* dari file-file yang disimpan di server dan entitas *requests* menampung sejumlah kata kunci yang didefinisikan oleh pengguna sistem, sedangkan entitas *english* dan *indonesian* adalah daftar kata-kata dalam Bahasa Inggris dan Bahasa Indonesia yang dapat dimasukkan sebagai kata kunci (*keyword*).



Gambar 3.17. ER Diagram Basis Data File Browser

Pada diagram entitas tersebut, tidak ada relasi antar entitas. Tiap entitas berdiri sendiri karena tidak ada keterkaitan.

3.2 Tipe Data dan Ukuran

Nama Atribut	Tipe Atribut	Ukuran Atribut
word	varchar	100

Tabel 3.7. Tabel Kosa Kata Bahasa Indonesia

Tabel 3.7 merupakan daftar atribut yang dimiliki oleh entitas *indonesian* tanpa kunci primer. Dalam tabel ini hanya terdapat atribut *word*

yang menampung seluruh kosa kata yang ada pada Bahasa Indonesia. Tipe *varchar* digunakan karena isi di dalam atribut *word* merupakan teks singkat dengan panjang maksimal atribut adalah 100 karakter.

Nama Atribut	Tipe Atribut	Ukuran Atribut
word	varchar	100

Tabel 3.8. Tabel Kosa Kata Bahasa Inggris

Tabel 3.8 merupakan daftar atribut yang dimiliki oleh entitas *english* tanpa kunci primer. Dalam tabel ini hanya terdapat atribut *word* yang menampung seluruh kosa kata yang ada pada Bahasa Inggris. Tipe *varchar* digunakan karena isi di dalam atribut *word* merupakan teks singkat dengan panjang maksimal atribut adalah 100 karakter.

Nama Atribut	Tipe Atribut	Ukuran Atribut
id	int	10
keyword	varchar	100
date_created	timestamp	

Tabel 3.9. Tabel Requests

Tabel 3.9 merupakan daftar atribut yang dimiliki oleh entitas *requests.* Dalam tabel ini terdapat atribut *id, keyword,* dan *date_created.* Penjelasan masing-masing atribut adalah sebagai berikut:

1. Kolom *id* merupakan kunci primer yang digunakan sebagai pengenal masing-masing isi tabel dengan tipe data *int* (integer) dan ukuran atribut maksimal 10 karakter.
2. Kolom *keyword* digunakan untuk menampung permintaan yang didefinisikan oleh *user.* Tipe atribut kolom *keyword* adalah *varchar* karena isinya merupakan teks singkat dengan ukuran maksimal atribut adalah 100 karakter.

3. Kolom *date_created* digunakan untuk menampung waktu saat kata kunci dibuat. Tipe ukuran *timestamp* digunakan agar waktu di-*generate* secara otomatis pada saat dilakukan *insert* data.

Nama Atribut	Tipe Atribut	Ukuran Atribut
id	int	10
name	varchar	100
title	varchar	100
description	longtext	
date_joined	timestamp	

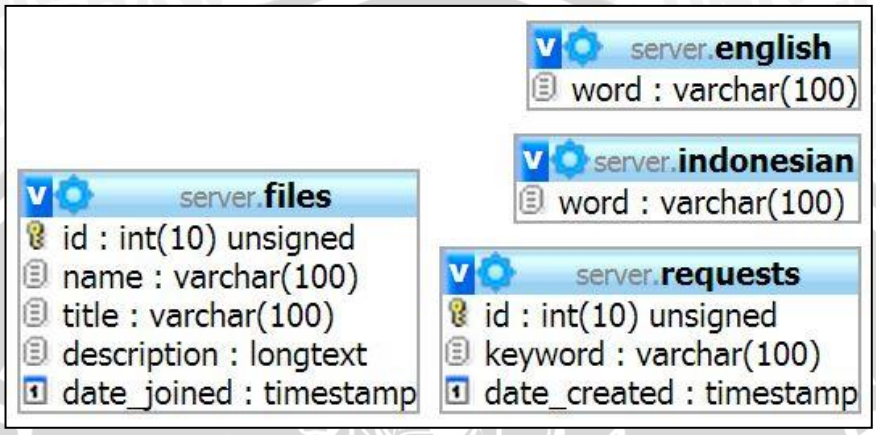
Tabel 3.10 Tabel Files

Tabel 3.10 merupakan daftar atribut yang dimiliki oleh entitas *files.* Dalam tabel ini terdapat atribut *id, name, title, description* dan *date_joined* yang isinya di-*generate* pada saat dilakukan *update* basis data. Penjelasan masing-masing atribut adalah sebagai berikut:

1. Kolom *id* merupakan kunci primer yang digunakan sebagai pengenal masing-masing isi tabel dengan tipe data *int* (*integer*) dan ukuran atribut maksimal 10 karakter.
2. Kolom *name* digunakan untuk menampung nama file saat dilakukan pemasukan data. Tipe atribut kolom *name* adalah *varchar* karena isinya merupakan teks singkat dengan ukuran maksimal atribut adalah 100 karakter.
3. Kolom *title* digunakan untuk menampung *metadata* nama file saat dilakukan pemasukan data. Apabila file tidak mengandung *title*, maka kolom diisi dengan nama “Untitled” secara otomatis. Tipe atribut kolom *title* adalah *varchar* karena isinya merupakan teks singkat dengan ukuran maksimal atribut adalah 100 karakter.
4. Kolom *description* digunakan untuk menampung isi file saat dilakukan pemasukan data. Tipe atribut kolom *description* adalah *longtext* karena isinya merupakan teks panjang.

5. Kolom *date_joined* digunakan untuk menampung waktu saat tabel diisi. Tipe ukuran *timestamp* digunakan agar waktu di-*generate* secara otomatis pada saat dilakukan *insert* data.

3.3 Struktur Data



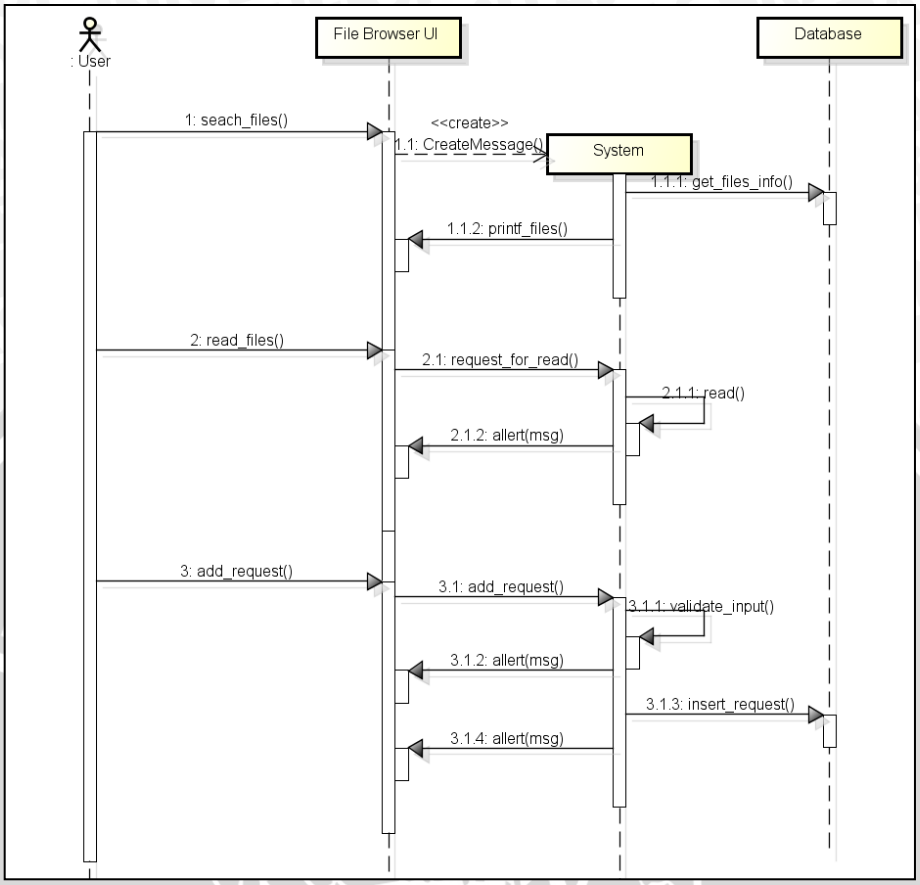
Gambar 3.18. Struktur Data

Gambar 3.18 merupakan tabel struktur data implementasi dari *Entity Diagram* yang ditunjukkan pada Gambar 3.17. Implementasi tiap entitas dibuat dalam bentuk tabel yang berisi nama tabel dan atribut-atributnya sesuai dengan Tabel 3.7 - Tabel 3.10.

4 Sequence Diagram

Sequence Diagram menunjukan apa saja aksi yang dapat dilakukan aktor pada suatu fungsi secara berurutan pada tiap aplikasi. Diagram ini hanya menampilkan urutan aksi per fungsi dan tidak menunjukkan kaitannya dengan fungsi lain.

➢ ***User***



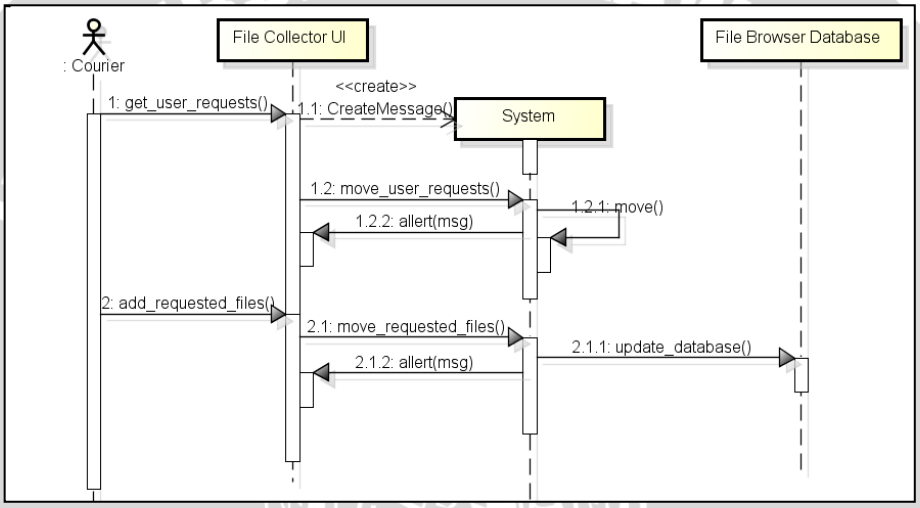
Gambar 3.19. Sequence Diagram: *User*

Sequence Diagram untuk *user* menunjukkan tiga hal yang dapat dilakukan, yaitu *search_file()*, *read_file()*, dan *add_request()*. Berikut ini adalah penjelasan masing-masing interaksi yang dilakukan.

- ***search_file()***
  - *User* meminta sistem untuk menampilkan file-file sesuai keinginannya.
  - Sistem menampilkan daftar file-file dalam basis data sesuai kata kunci masukan *user*.
- ***read_file()***
  - Sistem menampilkan file-file sesuai kata kunci masukan oleh *user*.
  - *User* memilih salah satu file.
  - Sistem membuka file yang telah dipilih oleh *user*.
  - Sistem menampilkan pesan *error* bila terjadi kesalahan.

- ***add_request()***
  - *User* menambahkan kata kunci permintaan kepada sistem.
  - Sistem melakukan validasi kata kunci masukan.
  - Sistem menampilkan pesan *error* bila masukan tidak valid.
  - Apabila masukan valid, sistem akan menyimpan kata kunci ke basis data.
  - Sistem menampilkan pesan *error* bila terjadi kesalahan.

➢ **Kurir Data di Sink POA**



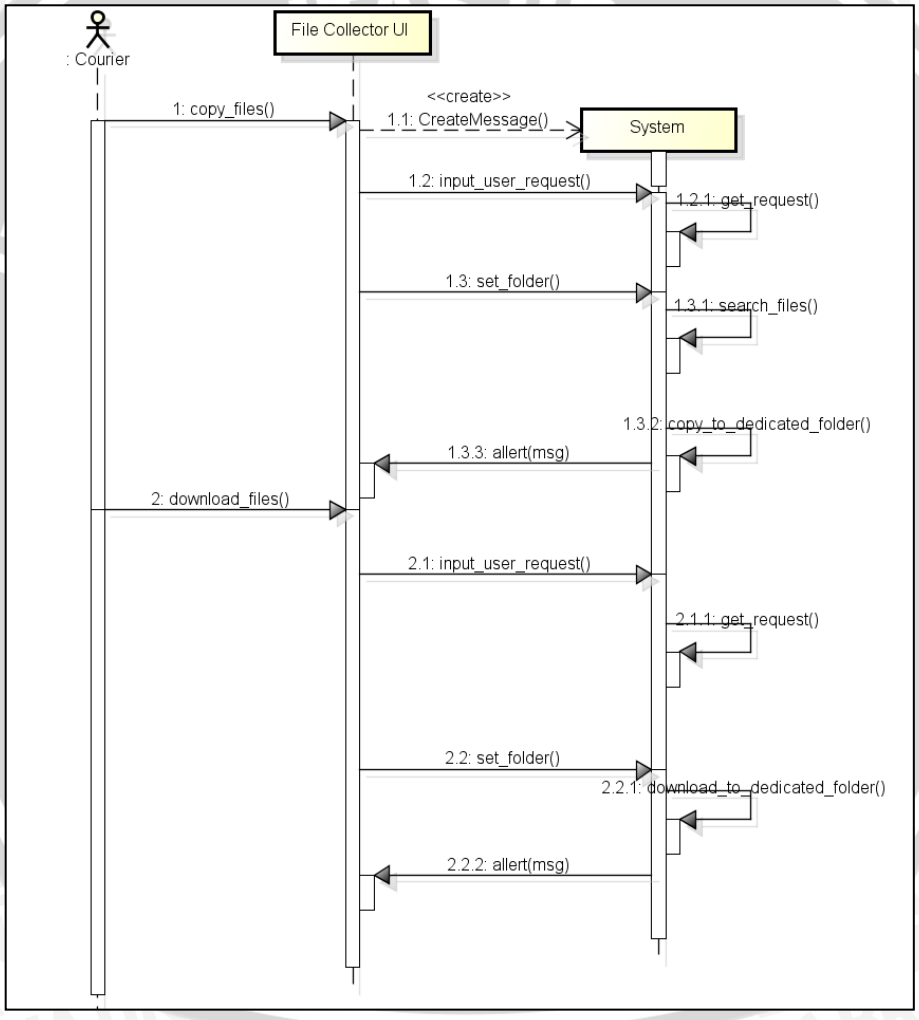
Gambar 3.20. Sequence Diagram: Kurir Data di Sink POA

Sequence Diagram untuk kurir data di Sink POA pada Gambar 3.20 menunjukkan dua hal yang dapat dilakukan, yaitu *get_user_requests()*, dan *add_requested_files()*. Berikut ini adalah penjelasan masing-masing interaksi yang dilakukan.

➢ ***get_user_requests()***
  - Kurir data meminta sistem untuk memindahkan file-file permintaan dari *local web server* ke dalam folder khusus yang ditentukan oleh program yang dibawa kurir data.
  - Sistem akan menampilkan pesan *error* bila terjadi kesalahan.

- ***add_requested_file()***
  - Kurir data meminta sistem untuk memindahkan semua file hasil pencarian yang ada dalam media penyimpanan ke *local web server*.
  - Sistem memindahkan file-file dan melakukan pembaharuan terhadap basis data.
  - Sistem menampilkan *error* bila terjadi kesalahan.

- **Kurir Data di Source POA**



Gambar 3.21. Sequence Diagram: Kurir Data di Source POA

Sequence Diagram untuk kurir data di Source POA pada Gambar 3.21 menunjukkan dua hal yang dapat dilakukan, yaitu *copy_files()*, dan

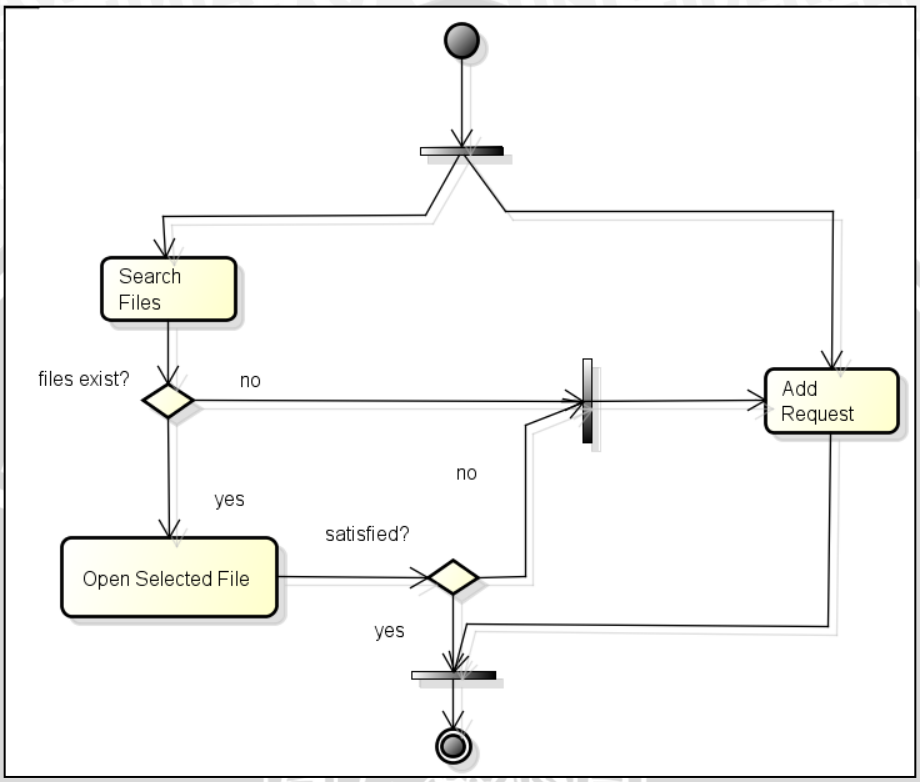
download_files(). Berikut ini adalah penjelasan masing-masing interaksi yang dilakukan.

- ***copy_files()***
  - Kurir data meminta sistem mengambil semua informasi pada file-file permintaan.
  - Sistem mengumpulkan semua permintaan dan menentukan folder tujuan penyalinan.
  - Sistem melakukan pencarian file-file sesuai kumpulan permintaan.
  - Sistem menyalin file-file sesuai permintaan ke dalam folder.
  - Sistem akan menampilkan pesan *error* bila terjadi kesalahan.
- ***download_files()***
  - Kurir data meminta sistem mengambil semua informasi pada file-file permintaan.
  - Sistem mengumpulkan semua permintaan dan menentukan folder tujuan penyalinan.
  - Sistem melakukan pencarian file-file sesuai kumpulan permintaan.
  - Sistem mengunduh file-file sesuai permintaan ke dalam folder.
  - Sistem akan menampilkan pesan *error* bila terjadi kesalahan.

5 Activity Diagram

Activity Diagram menunjukan alur penggunaan aplikasi, atau dengan kata lain, bagaimana tingkah laku sistem ketika suatu aksi diberikan sebagai masukan.

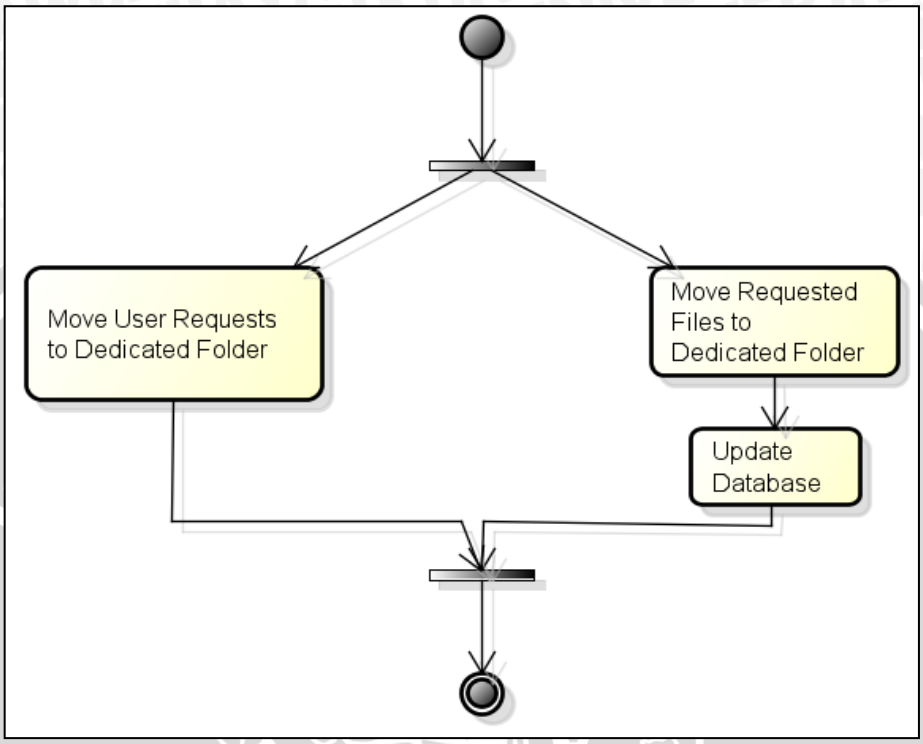
- ***User***



Gambar 3.22. Activity Diagram: *User*

Gambar 3.22 menunjukkan bahwa ketika *user* memasukkan kata kunci untuk melakukan pencarian file, sistem akan memeriksa apakah kata kunci cocok dengan isi file-file yang tersedia. Bila ada, maka file-file yang cocok ditampilkan, kemudian *user* dapat membuka file sesuai pilihan. Apabila tidak ada file yang ditemukan atau hasil kurang memuaskan, maka *user* dapat melakukan permintaan kepada sistem.

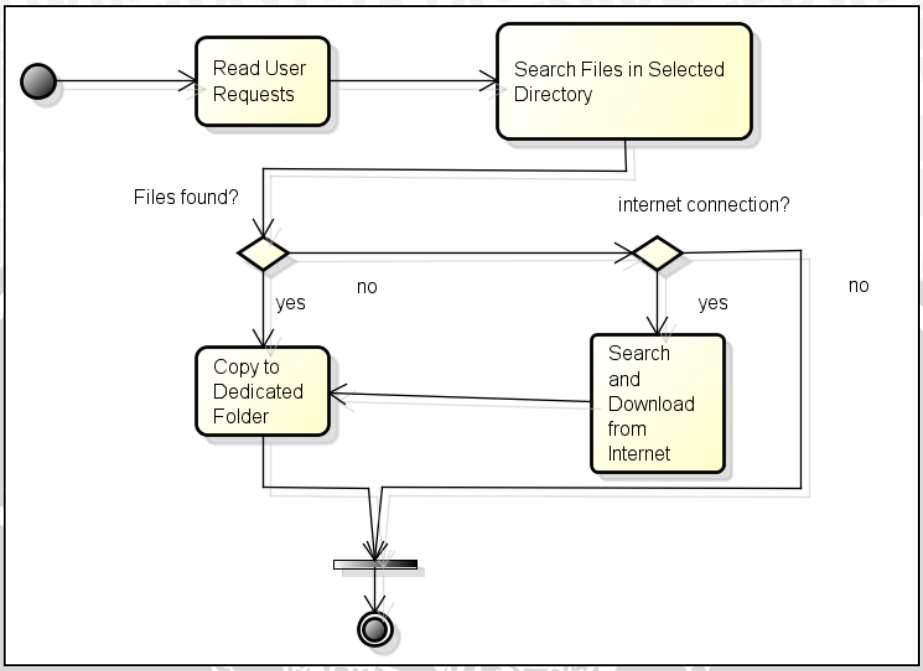
- **Kurir Data di Sink POA**



Gambar 3.23. Activity Diagram: Kurir Data di Sink POA

Gambar 3.23 menunjukkan bahwa ketika berada di Sink POA, kurir data dapat melakukan 2 hal. Pertama, memindahkan permintaan *user* yang sudah tersedia dalam bentuk file XML ke sebuah folder khusus dalam media penyimpanan. Kedua, memindahkan file-file yang sudah didapatkan sebelumnya di sumber data ke sebuah folder khusus pada server lokal, kemudian program secara otomatis akan meng-*update* basis data server.

➢ **Kurir Data di Source POA**



Gambar 3.24. Activity Diagram: Kurir Data di Source POA

Gambar 3.24 menunjukkan bahwa ketika berada di Source POA, kurir data menjalankan program yang melakukan pembacaan terhadap file-file permintaan dari *user* sebelumnya. Berdasarkan informasi yang tersedia, program melakukan pencarian file-file pada sebuah folder yang dipilih oleh kurir data. Apabila file ditemukan, maka file tersebut akan disalin ke sebuah folder khusus. Apabila file tidak ditemukan, maka kurir data dapat menentukan apakah file akan dicari di internet atau tidak. Ketika file yang dicari di internet ditemukan, maka program juga akan mengunduh file tersebut dan dipindahkan ke folder khusus.