

OPTIMALISASI MANAJEMEN PENERIMAAN & PENYIMPANAN DOKUMEN FAX MENGGUNAKAN ALGORITMA GREEDY

Intan Purnamasari¹, Oman Komarudin²

^{1,2}Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang
Email: ¹intan92@staff.unsika.ac.id, ²oman.komarudin@staff.unsika.ac.id

(Naskah masuk: 01 Agustus 2018, diterima untuk diterbitkan: 07 November 2019)

Abstrak

Mesin *fax* pada dasarnya mutlak diperlukan bagi sebuah instansi maupun organisasi tertentu. Namun kesulitan yang dihadapi oleh beberapa perusahaan kecil maupun menengah adalah tidak dapat mengalokasikan dana bagi pembelian mesin *fax*, perawatan, dan perlengkapan komputer sebagai media penunjang. Bagi perusahaan yang telah memiliki mesin *fax* pun, manajemen dokumen yang diterima melalui *fax* menjadi kesulitan tersendiri, dari mulai perawatan mesin *fax*, pemeriksaan ketersediaan kertas dan *cartridge* tinta, pemeriksaan dokumen yang diterima, serta penyimpanan arsip *fax* yang diterima. Metodologi penelitian yang akan digunakan pada penelitian ini adalah metode rakayasa perangkat lunak atau *Software Development Life Cycle* (SDLC) dengan model pemrograman cepat atau *Extreme Programming*. Desain pemodelan aplikasi menggunakan UML (*Unified Modelling Language*) yang diimplementasikan menggunakan bahasa pemrograman PHP sebagai algoritma program dan bahasa pemrograman HTML untuk desain tampilan aplikasi dengan *tools* aplikasi Notepad++. Hasil implementasi dari aplikasi ini adalah yang semula awalnya penerimaan dokumen *fax* dengan menggunakan mesin *fax* harus tercetak dan membutuhkan media kertas sebanyak 100% mengingat mesin *fax* hanya dapat menampilkan pesan dan dapat dibaca setelah tercetak namun setelah menggunakan aplikasi *digifax* ternyata dapat menghemat penggunaan kertas sebanyak 24,6% sehingga penggunaan kertas menurun menjadi 78,6%. Hal ini karena dengan aplikasi *digifax* dokumen atau *file fax* yang diterima dapat disimpan dan dapat dicetak (*print*) maupun diatur sesuai kebutuhan pengguna.

Kata kunci: *Dokumen Fax, Penerimaan Dokumen Fax, Algoritma Greedy*

OPTIMIZATION OF THE MANAGEMENT OF THE RECEPTION & STORAGE FAX DOCUMENT USING GREEDY ALGORITHMS

Abstract

Fax machine basically is absolutely necessary for a particular institution or organization. However, the difficulties faced by some small and medium enterprises are not able to allocate funds for the purchase of a fax machine, maintenance, and computer equipment as supporting media. For companies that already have a fax machine too, management documents received via fax into its own difficulties, ranging from fax machine maintenance, checking the availability of paper and ink cartridges, examination of documents received, and the received fax archival storage. The research methodology that will be used in research is a method of software engineering or Software Development Life Cycle (SDLC) with fast programming model or Extreme Programming. Design application modeling using UML (Unified Modeling Language) which is implemented using the PHP programming language as the program algorithm and HTML programming language for interface design tool applications with Notepad ++ applications. The results of the implementation of this application is that the original fax document acceptance initially using fax machines need to be printed and paper media as much as 100% given the fax machine can only display a message and can be read as printed digifax but after using the same practice was able to save the use of paper as much as 24, 6%, so the use of paper decreased to 78.6%. This is because the application digifax documents or files received faxes can be stored and can be printed (print) and arranged according to user needs.

Keywords: *Fax machine, Acceptance Fax Document, Greedy Algorithm*

1. PENDAHULUAN

Teknologi Informasi (TI) merupakan suatu item yang bermacam-macam dalam kemampuan yang digunakan dalam pembuatan, penyimpanan dan penyebaran data maupun informasi (Senn, 2000). Namun dalam berkomunikasi mapun memperoleh informasi membutuhkan sebuah media komunikasi berupa teknologi informasi yang dapat dimanfaatkan dalam berkomunikasi maupun memperoleh sebuah informasi (Sutarman, 2009). Salah satu teknologi informasi adalah mesin faksimili.

Faksimili adalah teknologi telekomunikasi yang digunakan untuk menyalin dan mengirimkan dokumen yang serupa dengan aslinya, menggunakan jaringan telepon ke mesin *fax* penerima yang kemudian bisa dicetak dikertas (Internet: 27 April 2013). Seluruh penerimaan pesan yang dikirim melalui *fax* akan tersimpan dalam *folder inbox* (kotak masuk). *Folder inbox* dapat diakses oleh pengguna yang mempunyai akun sebagai akses penerimaan dokumen *fax*. Namun pada saat *folder inbox* terisi penuh akan menyebabkan penumpukan dokumen *fax* dan tidak ada optimasi pilihan apakah dokumen *fax* akan dipindahkan atau dihapus. Akibatnya pemeriksaan rutin dokumen *fax* harus dilakukan agar tidak terjadi penumpukan dokumen *fax*. Hal ini membuat persoalan optimasi yang timbul bagaimana agar dokumen *fax* tidak menumpuk pada *folder inbox*.

Optimasi atau optimalisasi itu sendiri merupakan metode pemecahan masalah maksimalisasi atau minimalisasi. Menurut Kamus Besar Bahasa Indonesia Optimalisasi berasal dari kata dasar Optimal yang berarti terbaik atau dapat dikatakan sebagai sesuatu yang paling baik dari beberapa pilihan. Optimasi sangat bermanfaat untuk meningkatkan produktivitas kerja (Ambarwari A & Witdi N, 2016). Salah satu solusi optimasi dapat diatasi menggunakan sebuah algoritma, yaitu algoritma *greedy*. Algoritma *greedy* merupakan algoritma yang digunakan untuk pemecahan masalah dalam memilih sebuah solusi optimum dari beberapa elemen kandidat yang dapat digunakan untuk memecahkan masalah yang dimaksud (Munir, 2004). Algoritma ini akan digunakan untuk menentukan solusi terbaik berdasarkan *step by step* yang digunakan dalam menentukan optimasi sebagai pilihan solusi terbaik (Malik A., Sharma A. & Saroha V. 2013) terhadap dokumen *fax* yang akan dipindahkan pada *folder* tertentu maupun dihapus pada saat *folder inbox* terisi penuh oleh dokumen *fax* yang sudah diterima.

Penelitian sebelumnya yang terkait mengenai masalah optimasi dengan algoritma *Greedy* adalah pencarian *route* hotel dengan melakukan penyesuaian penggunaan algoritma berdasarkan karakteristik perjalanan yang dilakukan wisatawan (Herly M.A, Raharjana K.I, & Purbandini, 2015). Kemudian pada penelitian sebelumnya juga, algoritma *Greedy* dikatakan dapat melakukan

klasifikasi, konsistensi, tingkat konvergensi, dan adaptivitas terhadap beberapa persoalan seperti *inventory system* dan lain sebagainya (Mannor, Meir, & Zhang, 2003). Selain sebagai optimasi ternyata pada penelitian selanjutnya algoritma *greedy* dapat digunakan dapat diterapkan dalam pendekteksian aktifitas lansia pada saat berdiri, berjalan, duduk dan berbaring dengan menggunakan alat bantu sensor Arduino Mega. Penelitian tersebut bertujuan untuk menjaga aktivitas lansia agar dapat selalu terkontrol dan dipantau oleh orang disekitarnya (Wahyuningsih, 2018).

Berdasarkan permasalahan yang muncul, penelitian ini bermaksud untuk menggali konsep dan mengimplementasikan penggunaan aplikasi berbasis komputer untuk penerimaan dan manajemen dokumen *fax*. Selain untuk mendukung langkah *go green* yang bertujuan untuk mengurangi *global warming* atas dampak negatif penggunaan kertas yang tidak terpakai, dengan manajemen *file* melalui aplikasi yang dibangun, dapat dibuat sistematis penyimpanan *file* berdasarkan tanggal, hak akses, penerusan kepada tujuan yang dimaksud (disposisi), serta akan dibuat suatu optimasi sebagai solusi terbaik yang dilakukan pada saat *folder* penyimpanan yang terisi penuh dokumen *fax*.

2. METODE

Metodologi penelitian menggunakan *System Life Cycle* (SDLC) model pengembangan pemrograman cepat (*Extreme Programming* atau XP) (Pressman, 2010). XP merupakan salah satu pendekatan pengembangan perangkat lunak yang banyak digunakan dalam perangkat lunak cepat dan sedikit lebih rinci (Beck, 2002). XP pada dasarnya menggunakan suatu pendekatan berorientasi objek (Object Oriented) sebagai suatu paradigma pengembangan yang diinginkan yang didalamnya mencakup sebuah aturan dan praktik-praktik yang terjadi dalam konteks empat kegiatan kerangka kerja, yaitu: perencanaan, perancangan, pengkodean, dan pengujian (Pressman, 2010). Tahapan penelitian yang akan dilakukan yaitu : 1) Perencanaan (*User Stories*), 2) Perancangan, 3) Pengkodean, 4) Pengujian.

2.1. Perencanaan (*User Stories*)

User stories pada penelitian ini merupakan kumpulan cerita awal pengguna yang masuk kedalam klasifikasi pengguna mesin *fax* mengenai penggunaan dan pemanfaatan mesin *fax* itu sendiri bagi pengguna. Kemudian pengguna tersebut akan menjadi calon pengguna terhadap sistem yang akan dibuat pada perangkat lunak yang akan dibangun. Hasil dari analisis penggunaan mesin *fax* ini akan menjadi sebuah kumpulan cerita atau disebut dengan *user stories* yang dikemukakan oleh pengguna (*user*) yang dibuat sebuah ringkasan cerita yang dijadikan

sebagai perencanaan awal bagi penulis dalam membangun aplikasi.

1. Analisis Kebutuhan Fungsional

Analisis ini dilakukan untuk mengetahui secara langsung mengenai kebutuhan pengguna (*user*) terhadap pemanfaatan mesin *fax* yang lebih maksimal. Analisis ini bertujuan untuk memudahkan pengembang perangkat lunak dalam membuat fungsionalitas tambahan pada perangkat lunak yang akan dibangun sesuai dengan keperluan penggunaannya (*user*).

2. Analisis Pengguna Sistem

Analisis pengguna sistem ini merupakan analisis pengguna yang didapat berdasarkan klasifikasi pengguna mesin *fax* yang menjadi calon pengguna sistem yang akan dibuat pada perangkat lunak yang akan dibangun. Analisis ini bertujuan untuk memudahkan pengembang perangkat lunak dalam membangun sistem yang akan dibuat sesuai kebutuhan pengguna (*user*) pada masalah yang ada saat ini.

2.2. Perancangan

Sistem ini dirancang agar mempermudah sebagian orang dalam mengatur penerimaan pesan melalui mesin *fax*, dan perancangan perangkat lunak yang akan dibangun menggunakan pemodelan *Unified Modeling Language* (UML). Adapun perancangan yang akan dipakai dalam sistem ini hanya menggunakan *Class Responsibility Collaboration* (CRC) dan menggunakan beberapa pemodelan UML, yaitu sebagai berikut:

1. *Class Responsibility Collaboration* (CRC) Card untuk memperhatikan alur sistem secara keseluruhan dengan pembagian kelas-kelas yang menunjukkan kebutuhan-kebutuhan dan tanggung jawab antara kelas yang dibuat sesuai dengan atribut yang digunakan.
2. *Use Case* yang digunakan sebagai alur aktivitas pengguna (*user*) terhadap perangkat lunak yang akan dibangun.
3. *Activity* diagram yang merupakan perluasan dari *Use Case* yang akan menghasilkan sebuah interaksi antara pengguna dengan perangkat lunak yang kemudian akan penulis terjemahkan menjadi sebuah *User Interface* atau antar muka pengguna pada perangkat lunak yang akan dibangun.

2.3. Pengkodean

Pada tahap ini penulis mengimplementasikan seluruh hasil dari desain arsitektur aplikasi, desain *database*, dan desain *interface* yang sebelumnya telah dibuat ke dalam bentuk yang sesungguhnya. *Tool* atau perangkat

lunak yang akan digunakan pada dasarnya adalah DBMS MySQL sebagai implementasi dari desain *database*, *script* atau *source code* bahasa pemrograman PHP yang akan digunakan untuk pembuatan algoritma program dan menggunakan desain tampilan web dengan menggunakan HTML.

Algoritme program akan disertai dengan penggunaan algoritma *Greedy* (Munir, R. 2004) dalam optimasi solusi terbaik untuk memaksimalkan kapasitas suatu penyimpanan data yang digunakan untuk menyimpan dokumen *fax* nantinya. Adapun pemaparan skema umum algoritma *greedy* dalam *pseudocode* sebagai berikut:

```
function greedy(input C:
himpunan_kandidat)→ himpunan_kandidat
{ Mengembalikan solusi dari persoalan
optimasi dengan algoritma greedy
Masukan: himpunan_kandidat C Keluaran:
himpunan_solusi yang bertipe
himpunan_kandidat
}

Deklarasi:
x : kandidat
S : himpunan_kandidat

Algoritma:
S ← {} { inisialisasi S dengan kosong }
while (not SOLUSI(S)) and (C ≠ {} ) do
  x ← SELEKSI(C) { pilih sebuah kandidat
dari C }
  C ← C - {x} { elemen himpunan_kandidat
berkurang satu }
  if LAYAK(S U {x}) then
    S ← S U {x}
  endif
endwhile
{SOLUSI(S) or C = {} }
if SOLUSI(S) then
  return S
else
  write('tidak ada solusi')
endif
```

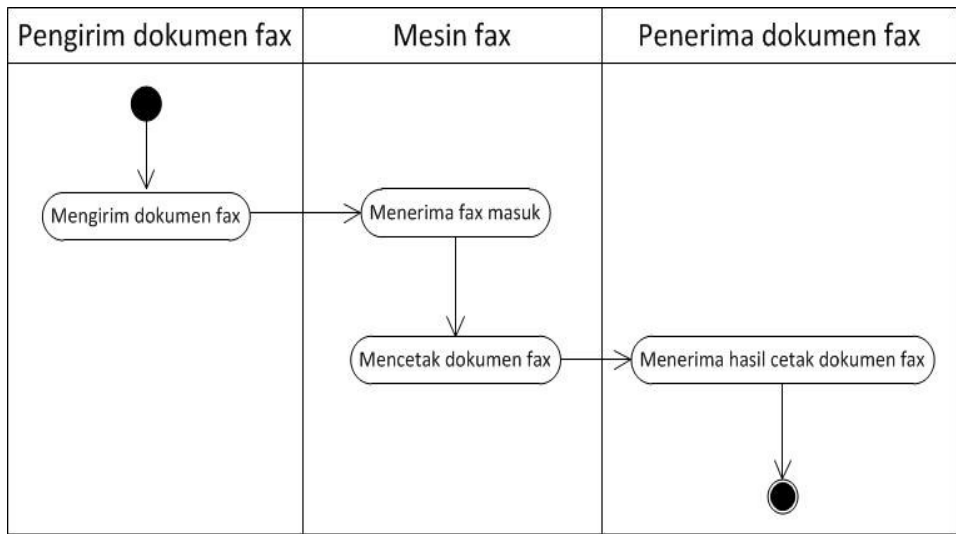
Gambar 1. Skema Umum Algoritma *Greedy*

2.4. Pengujian

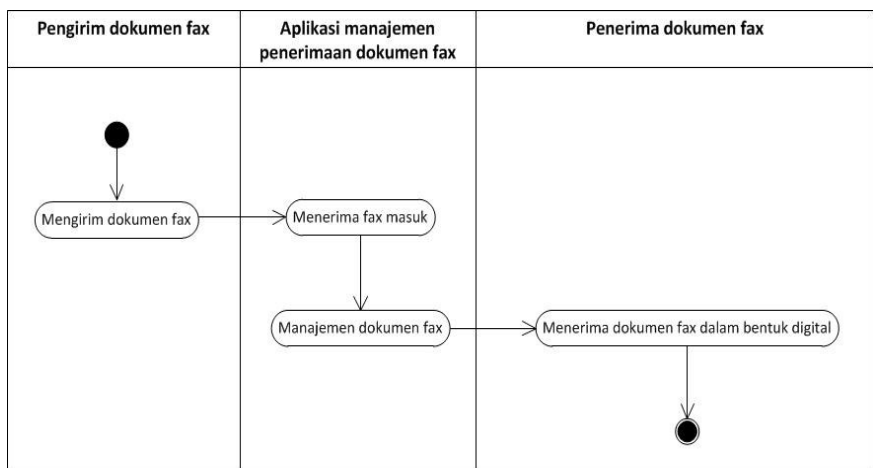
Tahap ini dilakukan pada saat keseluruhan tahap sebelumnya telah dilakukan. Tahap pengujian yang akan dilakukan ada dua jenis yaitu dengan menggunakan *White Box Testing* dan *Black Box Testing*.

3. HASIL PENELITIAN

Berdasarkan dari hasil wawancara dan observasi penulis mendapatkan data terkait dengan penggunaan mesin *fax* terhadap beberapa organisasi dan instansi tertentu maupun perorangan.



Gambar 2. Activity Diagram Penerimaan Dokumen fax dengan Menggunakan Mesin fax



Gambar 3. Activity Diagram Sistem yang Diusulkan

Melihat dari sistem penerimaan dokumen fax kebanyakan pada saat ini, dengan tidak adanya manajemen penerimaan dokumen fax ditambah lagi dokumen fax tersebut tidak tersimpan. Maka diusulkan sistem penerimaan dokumen fax melalui mesin fax dengan manajemen penerimaan dokumen fax dalam bentuk activity diagram pada Gambar 3

Aplikasi yang telah dibangun diberi nama digifax, pembangunan digifax menggunakan tahapan XP ini mengalami hingga lebih dari satu kali iterasi. Iterasi dilakukan pada saat tahapan perancangan dan pengkodean dilakukan demi mendapat hasil yang maksimal.

3.1. User Stories

Berdasarkan dari hasil analisis yang telah dilakukan terhadap beberapa pengguna mesin fax menjadi sebuah cerita awal (user stories) dan didapat beberapa pendefinisian masalah sebagai berikut:

1. Dokumen fax yang diterima harus terlebih dahulu tercetak agar pesan yang diterima dapat terbaca.

2. Kebanyakan dari pengguna mesin fax kesulitan dalam mencari dokumen fax yang sudah diterima sebelumnya karena dokumen fax yang telah diterima tidak tersimpan.
3. Dokumen fax yang diterima dapat diakses oleh siapa pun sehingga tidak ada keamanan hak akses dapat dikatakan tidak terjamin.
4. Dokumen fax yang diterima tidak selalu merupakan dokumen yang dianggap penting oleh beberapa pengguna mesin fax, sehingga ada banyak beberapa dokumen fax yang terbuang begitu saja.
5. Dokumen fax yang terbuang begitu saja membuat sumber daya seperti kertas dan cartridge tinta menjadi terbuang percuma.

Analisis kebutuhan dilakukan untuk mengetahui secara jelas dan tepat terhadap masalah yang ada saat ini yang dihadapi oleh pengguna khususnya bagi pengguna mesin fax.

1. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui kebutuhan apa saja yang dibutuhkan oleh pengguna (*user*) itu sendiri terhadap pemanfaatan mesin *fax*. Dari hasil wawancara dan observasi sesuai dengan *user stories* yang telah dibuat, maka didapat beberapa kebutuhan pengguna terhadap pemanfaatan mesin *fax* yaitu:

- a. Penerimaan dokumen *fax* yang terkomputerisasi sehingga dokumen *fax* yang diterima tidak harus tercetak terlebih dahulu agar pesan didalamnya dapat terbaca. Sehingga hal ini juga dapat menghemat penggunaan kertas dan *cartridge* tinta yang pada umumnya dibutuhkan oleh mesin *fax*.
 - b. Manajemen penerimaan dokumen *fax* yang teratur berdasarkan tanggal penerimaan dokumen *fax* dan dapat tersimpan, sehingga memudahkan pencarian dokumen *fax* pada saat dibutuhkan kembali.
 - c. Penerapan sistem pengelolaan hak akses terhadap pengguna yang berkepentingan saja yang dapat menerima dokumen *fax*.
2. Analisis Pengguna Sistem

Analisis pengguna sistem dilakukan untuk mencari pengguna yang tepat terhadap aplikasi yang akan dibangun. Hasil analisis ini didapat dari hasil analisis terhadap klasifikasi pengguna mesin *fax* yang telah dilakukan sebelumnya hanya ada dua pengguna sistem pada aplikasi ini yaitu admin dan *user*.

3.2. Perancangan

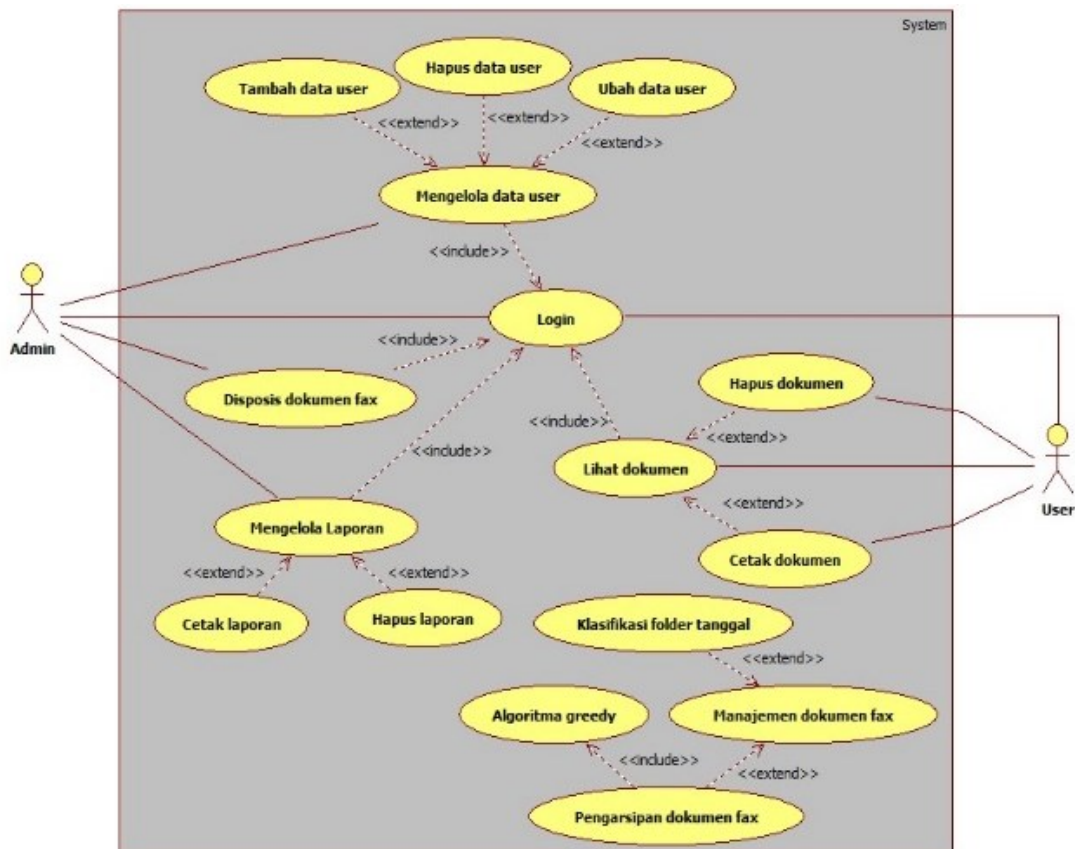
Perancangan dilakukan untuk pemetaan atau gambaran desain aplikasi yang akan dibangun. Desain aplikasi yang akan dibangun menggunakan perancangan dengan *Class Responsibility Diagram* (CRC) *Card* dan beberapa perancangan dengan pemodelan UML. Dimana desain arsitektur aplikasi dibuat menggunakan CRC *Card*, *use case diagram*, *activity diagram*, dan *sequence diagram*. Untuk desain *database* menggunakan *class diagram* dan untuk representasi dari desain *interface* menggunakan *tools* Dia. Berikut perancangan secara garis besar sistem yang telah dibangun.

3.3. Pengkodean

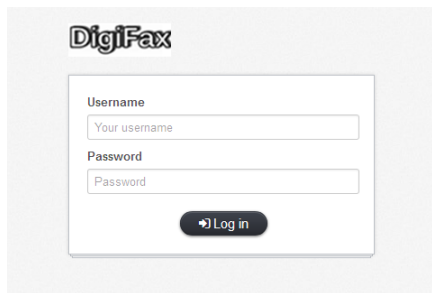
Implementasi antarmuka didapat dari desain *interface* atau tampilan dari aplikasi yang telah dibuat, berikut ini beberapa hasil implementasi dari desain *interface* atau desain antarmuka yang telah dibuat:

a. Tampilan *formLogin*

Berikut merupakan hasil implementasi dari desain login yang telah dibuat pada aplikasi *digifax* :



Gambar 4. Use case Diagram Aplikasi Manajemen Penerimaan Dokumen Fax

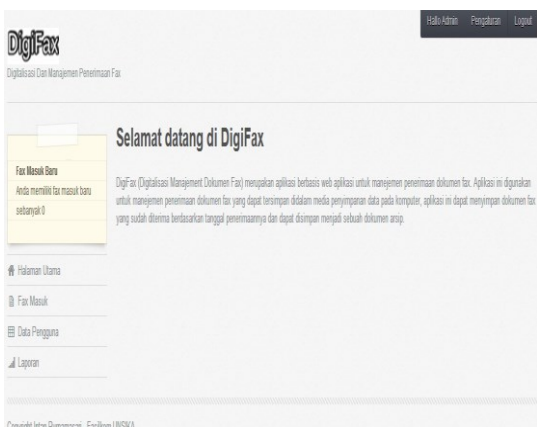


Gambar 5. Tampilan form login

Dari Gambar 5 diatas terdapat dua isian data yaitu *username* dan *password* yang akan diisi oleh admin maupun user sesuai dengan *username* dan *password* yang dimilikinya jika semua kolom isian teks terisi dengan benar maka tekan tombol bertuliskan *login* untuk masuk ke menu halaman utama.

- b. Tampilan form halaman utama (*home*) untuk admin

Berikut merupakan hasil implementasi dari desain halaman utama (*home*) bagi admin yang telah dibuat pada aplikasi *digifax*.



Gambar 6. Tampilan Form Halaman Utama (*home*) Admin

Gambar 6 merupakan tampilan utama bagi admin dari aplikasi *digifax* dimana terdapat beberapa menu pilihan pada samping kiri halaman yaitu menu *Halaman Utama* yang sedang ditampilkan disamping kanannya, kemudian ada menu *FaxMasuk* untuk melihat daftar dokumen *fax* yang telah diterima, *Data Pengguna* untuk akses *user* atau pengguna dan menu *Laporan* untuk melihat jumlah *fax* yang masuk.

- c. Tampilan form halaman utama (*home*) untuk user
- Berikut merupakan hasil implementasi dari desain halaman utama (*home*) bagi *user* yang telah dibuat pada aplikasi *digifax*.

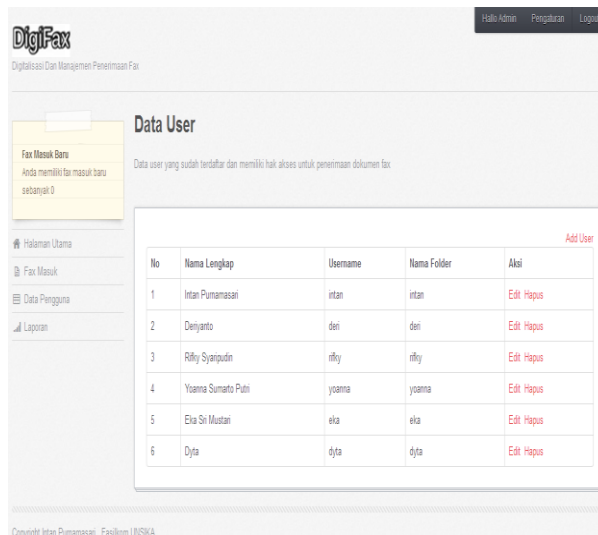
Gambar 7 merupakan tampilan utama bagi *user* dari aplikasi *digifax* dimana terdapat beberapa menu pilihan pada samping kiri halaman yaitu menu *Halaman Utama* yang sedang ditampilkan disamping kanannya dan ada menu *FaxMasuk*

untuk melihat daftar dokumen *fax* yang telah diterima.



Gambar 7. Tampilan Form Halaman Utama (*home*) User

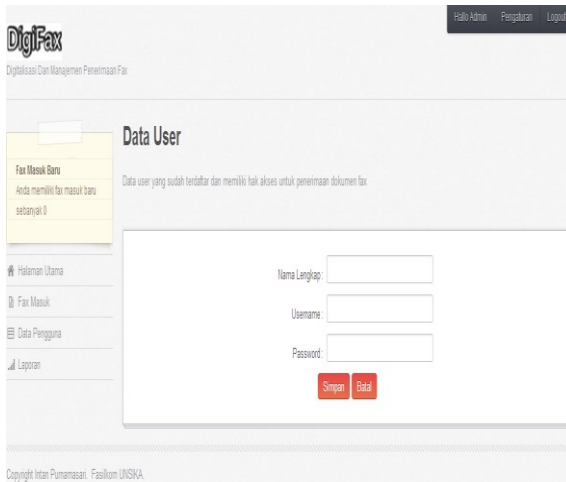
- d. Tampilan form lihat data *user*
- Berikut merupakan hasil implementasi dari desain halaman untuk menampilkan data *user* yang dapat diakses hanya oleh admin yang telah dibuat pada aplikasi *digifax*.



Gambar 8. Tampilan Form Lihat Data *user*

Gambar 8 menampilkan jumlah data *user* yang tersimpan didalam database yang dapat ditampilkan oleh admin dengan memilih menu *Data User* untuk membuka daftar data *user*, admin dapat memilih aksi *Edit* dan *Hapus* untuk memperbaharui data *user* yang sebelumnya sudah ditambahkan, selain itu admin dapat menambahkan data *user* yang baru dengan mengklik *Add User* agar tampil sebuah tampilan atau form untuk menambah data baru.

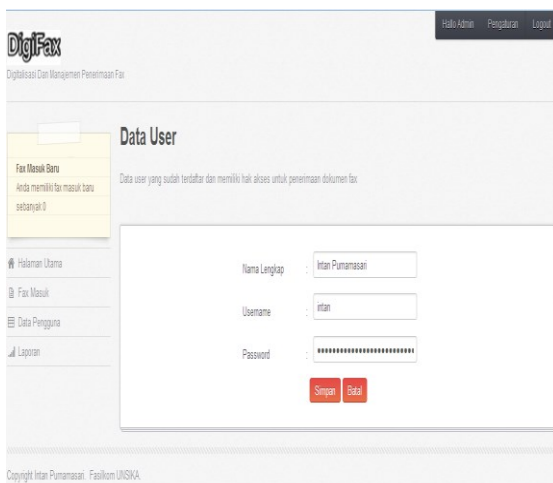
- e. Tampilan form tambah data *user*
- Berikut merupakan hasil implementasi dari desain halaman untuk menambah data *user* yang dapat diakses hanya oleh admin yang telah dibuat pada aplikasi *digifax* :



Gambar 9. Tampilan Form Tambah Data User

Gambar 9 merupakan tampilan atau *form* untuk menambah data *user* baru jika semua kolom isian terisi dengan lengkap maka jika admin menekan tombol *Simpan* data yang sudah dimasukkan akan tersimpan ke dalam *database*, sebaliknya jika admin menekan tombol *Batal* maka akan kembali ke menu sebelumnya.

- f. Tampilan *form* edit atau ubah data *user*
Berikut merupakan hasil implementasi dari desain halaman untuk mengubah data *user* yang dapat diakses hanya oleh admin yang telah dibuat pada aplikasi *digifax* :

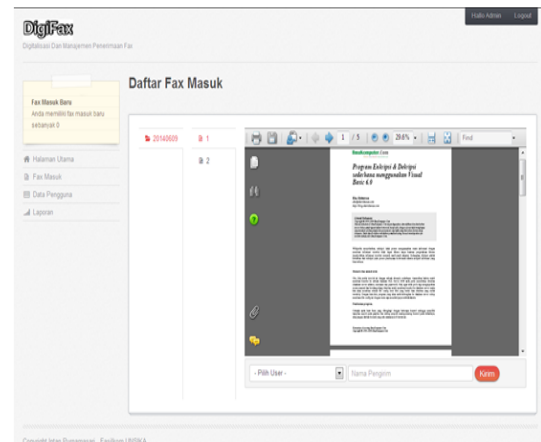


Gambar 10. Tampilan Form Edit atau Ubah Data user

Gambar 10 merupakan tampilan atau *form* untuk mengubah data *user* baru jika semua kolom isian terisi dengan lengkap maka jika admin menekan tombol *Simpan* data yang sudah dimasukkan atau sudah selesai diubah akan tersimpan ke dalam *database*, sebaliknya jika admin menekan tombol *Batal* maka akan kembali ke menu sebelumnya.

- g. Tampilan *form* disposisi dokumen *fax*
Berikut merupakan hasil implementasi dari desain halaman untuk disposisi dokumen *fax* yang dilakukan oleh admin untuk hak akses dokumen

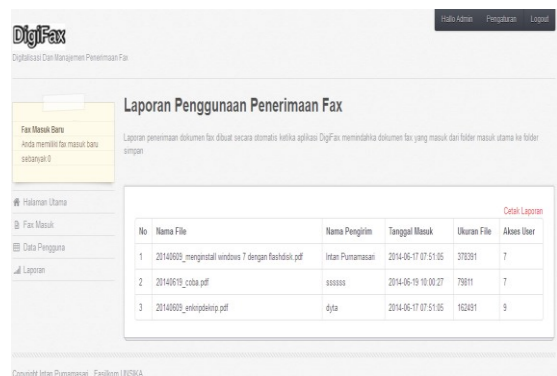
fax terhadap *user* yang telah dibuat pada aplikasi *digifax* :



Gambar 11. Tampilan Form Disposisi Dokumen *fax*

Gambar 11 merupakan tampilan atau *form* untuk disposisi dokumen *fax* kepada *user* sesuai dengan hak akses dokumen *fax* yang sudah diterima. Jika dokumen *fax* ditujukan untuk satu *user* admin memilih bagian *list* data *user* yang dimaksud kemudian mencantumkan nama pengirim lalu menekan tombol *Kirim*, maka dokumen *fax* yang sudah terkirim tersebut akan tersimpan ke dalam *folder* simpan *user* yang telah ditujukan.

- h. Tampilan *form* laporan penerimaan dokumen *fax*
Berikut merupakan hasil implementasi dari desain halaman laporan penerimaan dokumen *fax* yang telah dibuat pada aplikasi *digifax*:



Gambar 12. Tampilan Form Laporan Penerimaan Dokumen *fax*

Gambar 12 merupakan tampilan atau *form* untuk menampilkan laporan penerimaan dokumen *fax* yang sudah tersimpan didalam *database*, *form* ini hanya dapat diakses oleh admin dan apabila admin memilih aksi *CetakLaporan* maka laporan penerimaan dokumen *fax* akan tercetak dalam bentuk *hardcopy*.

3.4. Pengujian

Pengujian dilakukan dengan dua metode yaitu menggunakan *Black Box Testing* dan *White Box Testing*. *Black Box Testing* dilakukan untuk menguji fungsi-fungsi yang terdapat pada aplikasi apakah sudah sesuai atau belum sehingga dapat dikatakan *user friendly*. Sedangkan *White Box Testing* digunakan untuk pengujian terhadap logika program yang dibuat pada aplikasi *digifax*.

1. Pengujian menggunakan *Whitebox*

Pengujian *whitebox* dilakukan untuk menguji logika program dengan algoritma program terhadap aplikasi yang telah dibuat. Pengujian *whitebox* untuk klasifikasi *folder* berdasarkan tanggal terbagi atas dua opsi yaitu *admin* dan *user*, sebelum *admin* dan *user* memiliki *folder* berdasarkan tanggal sesuai dengan tanggal dokumen *fax* yang masuk, maka *admin* dan *user* harus terlebih dahulu memiliki *folder* simpan sebagai *folder* utama untuk menyimpan *folder-folder* baru beserta isi *file* dokumen *fax*-nya sesuai tanggal penerimaan dokumen *fax*. Berikut ini merupakan pengujian *whitebox* cek validasi untuk membuat *folder* simpan berdasarkan dua opsi yaitu bagi *admin* dan *user*.

```

<?php
session_start();
include_once('koneksi.php');
1 date_default_timezone_set
  ("Asia/Jakarta");
  $tgl = date('Ymd');

2 if($_SESSION['_loginID']=='2')
  {

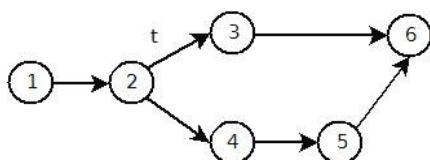
  $folderMasuk =
  option('folder_masuk');
3 $folderSimpan =
  option('folder_simpan');
  include 'greedy.php';

4 } else {

  $folderMasuk=option('folder_fax'
  )."/".

5 $_SESSION['_loginID'];
  $folderSimpan=$folderMasuk;

6 }
    
```



Gambar 13. Pengujian *Whitebox* untuk Validasi Membuat *Folder* Simpan bagi *Admin* atau *User*

$$\begin{aligned}
 V(G) &= E - N + 2 & V(G) &= P + 1 \\
 &= 6 - 6 + 2 & &= 1 + 1 \\
 &= 2 & &= 2
 \end{aligned}$$

Jadi, dapat disimpulkan dari pengujian *whitebox* pada pengujian diatas terdapat 2 alur path untuk

membuat *foldersimpan* sebagai penyimpanan dokumen *fax* yang telah diterima, yaitu sebagai berikut:

1. Path 1 = 1 – 2 – 3 – 6
2. Path 2 = 1 – 2 – 4 – 5 – 6

Selanjutnya setelah pengujian diatas dilakukan maka langkah selanjutnya adalah mengecek apakah *folder* yang baru dibuat sudah ada sebelumnya atau belum. Maka dilakukan pengujian *whitebox* cek validasi untuk mengecek apakah *folder* yang akan dibuat sudah ada atau belum sebelumnya seperti gambar 14.

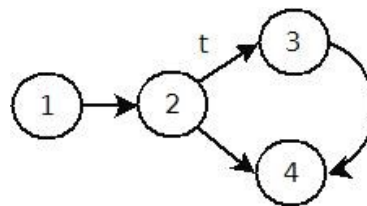
```

<?php
1 $folderFax=$folderSimpan."/".$tg
  l;

2 if(!file_exists($folderSimpan
3 {

  mkdir($folderSimpan);

4 }
  ?>
    
```



Gambar 14. Pengujian *Whitebox* Validasi untuk Membuat *Folder* Simpan

$$\begin{aligned}
 V(G) &= E - N + 2 & V(G) &= P + 1 \\
 &= 4 - 4 + 2 & &= 1 + 1 \\
 &= 2 & &= 2
 \end{aligned}$$

Jadi, dapat disimpulkan dari pengujian *whitebox* pada pengujian diatas terdapat 2 alur path untuk mengecek apakah *folder* yang akan dibuat sudah ada atau belum sebelumnya sebagai penyimpanan dokumen *fax* yang telah diterima, yaitu sebagai berikut:

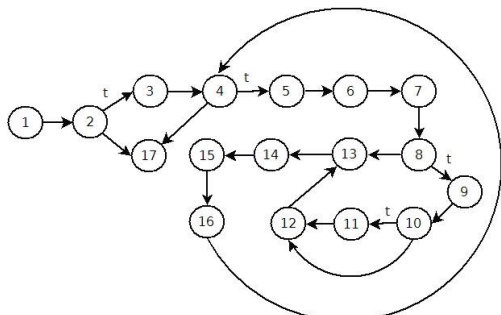
1. Path 1 = 1 – 2 – 3 – 4
2. Path 2 = 1 – 2 – 4

Langkah selanjutnya setelah *folder* simpan untuk *admin* dan *user* telah dibuat maka *folder* masuk yang merupakan *folder* utama untuk menyimpan dokumen *fax* yang diterima pertama kali dicek oleh *admin* sehingga dokumen atau *file fax* yang telah diterima tersimpan pada *folder* simpan milik *admin*, kemudian *user* mendapat dokumen atau *file fax* yang diterima dengan menunggu konfirmasi dari *admin*. Berikut pengujian *whitebox* untuk memindahkan dokumen atau *file fax* dari *folder* masuk ke *folder* simpan milik *admin* maupun *user*.


```

1 <?php
  $bukaFolderMasuk =
    opendir($folderMasuk);
2 if($bukaFolderMasuk)
3 {
4 while(FALSE != ($fileMasuk =
  readdir($bukaFolderMasuk)))
5 {
6 if($fileMasuk != "." &&
  $fileMasuk != "..")
  {
7 $tgl_file =
  filectime($folderMasuk."/".$
  fileMasuk); $tgl1 =
  date('Ymd', $tgl_file);
8 if($SESSION[' loginID ']== '2')
  {
  rename($folderMasuk."/".$fileMa
  suk,
  $folderMasuk."/".$tgl1."_".$
  fileMasuk);
9 $namaBaru =
  $tgl1."_".$fileMasuk;
  $folderFax =
  $folderSimpan."/".$tgl1;
10 if(!file_exists($folderFax))
  {
11 mkdir($folderFax);
  }
  $fileAsal =
  $folderMasuk."/".$namaBaru;
  $fileTujuan =
12 $folderFax."/".$namaBaru;
  copy($fileAsal, $fileTujuan);
  unlink($fileAsal);
13 }else{
  $namaBaru = $fileMasuk;
14 }
15 }
16 }
17 }
?>

```



Gambar 15. Pengujian *Whitebox* Memindahkan Dokumen *fax* dari *Folder* Utama ke *Folder* Simpan

$$\begin{aligned}
 V(G) &= E - N + 2 & V(G) &= P + 1 \\
 &= 20 - 17 + 2 & &= 4 + 1 \\
 &= 5 & &= 5
 \end{aligned}$$

Jadi, dapat disimpulkan dari pengujian *whitebox* pada pengujian di atas terdapat 5 alur path untuk memindahkan dokumen atau *filefax* dari *folder* masuk ke *folder* simpan milik admin yaitu sebagai berikut:

1. Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 4 – 17
2. Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16 – 4 – 17

3. Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16
4. Path 4 = 1 – 2 – 3 – 4 – 17
5. Path 5 = 1 – 2 – 17

2. Pengujian menggunakan *Blackbox*
 Pengujian *blackbox* dilakukan untuk menguji data masukkan maupun keluaran terhadap respon sistem yang dilakukan.

a. Pengujian *blackbox login*
 Berikut merupakan tabel penjelasan *blackbox* untuk *form login* yang telah dibuat:

Tabel 1. Pengujian *Blackbox Login*

Data Masukkan	Harapan	Respon sistem	Hasil
<i>Username</i> dan <i>Password</i> valid atau benar.	Tampil halaman utama (<i>home</i>)	Tampil halaman utama (<i>home</i>)	Diterima
<i>Username</i> dan <i>Password</i> kosong.	Tampil pesan “ <i>Password</i> belum diisi”	Tampil pesan “ <i>Password</i> belum diisi”	Diterima
<i>Username</i> diisi dan <i>Password</i> kosong.	Tampil pesan “ <i>Username</i> belum diisi”	Tampil pesan “ <i>Username</i> belum diisi”	Diterima
<i>Username</i> kosong dan <i>Password</i> diisi.	Tampil pesan “ <i>Username</i> dan <i>Password</i> belum diisi”	Tampil pesan “ <i>Username</i> dan <i>Password</i> belum diisi”	Diterima
<i>Username</i> dan <i>Password</i> tidak valid atau salah.	Tidak masuk ke halaman utama dan Tampil pesan “ <i>Username</i> tidak ditemukan !”	Tidak masuk ke halaman utama dan Tampil pesan “ <i>Username</i> tidak ditemukan !”	Diterima

b. Pengujian *blackbox* tambah data *user*
 Berikut merupakan tabel penjelasan *blackbox* untuk *form login* yang telah dibuat:

Tabel 2. Pengujian *Blackbox* Tambah Data *User*

Data Masukkan	Harapan	Respon sistem	Hasil
Semua isian diisi dengan benar	Data <i>user</i> tersimpan dalam <i>database</i> , dan tampil pesan data berhasil disimpan.	Data <i>user</i> tersimpan dalam <i>database</i> , dan tampil pesan data berhasil disimpan.	Diterima
Semua isian tidak terisi dengan benar	Data <i>user</i> tidak tersimpan dalam <i>database</i> , dan tampil pesan data gagal disimpan.	Data <i>user</i> tidak tersimpan dalam <i>database</i> , dan tampil pesan data gagal disimpan.”	Diterima

c. Pengujian *blackbox* disposisi dokumen *fax*
 Berikut merupakan tabel penjelasan *blackbox* untuk disposisi dokumen *fax* yang

dilakukan oleh admin dengan mengirimkan dokumen *fax* ke *folder user*:

Tabel 3. Pengujian *Blackbox* Disposisi Dokumen *fax*

Data Masukkan	Harapan	Respon sistem	Hasil
Mengirim dokumen <i>fax</i> pada data user yang benar dengan menekan tombol "Kirim"	Dokumen <i>fax</i> tersimpan pada <i>folder user</i> dan tampil pesan <i>fax</i> berhasil dikirim.	Dokumen <i>fax</i> tersimpan pada <i>folder user</i> dan tampil pesan <i>fax</i> berhasil dikirim.	Diterima
Tidak mengirim dokumen <i>fax</i> pada <i>user</i> dan tidak menekan tombol "Kirim"	Dokumen <i>fax</i> tidak tersimpan pada <i>folder user</i> dan tidak akan tampil pesan <i>fax</i> berhasil dikirim.	Dokumen <i>fax</i> tidak tersimpan pada <i>folder user</i> dan tidak akan tampil pesan <i>fax</i> berhasil dikirim.	Diterima

4. EVALUASI SISTEM

Hasil wawancara dan implementasi aplikasi selama satu bulan penerimaan dokumen *fax* yang harus tercetak berkurang setelah menggunakan aplikasi *digifax*. Berdasarkan data yang didapat *user* atau pengguna harus mencetak keseluruhan dokumen *fax* yang diterima berapa pun jumlahnya, namun setelah menerapkan aplikasi *digifax* dokumen *fax* yang diterima tidak harus dicetak secara keseluruhan. Sehingga penggunaan kertas untuk menerima dokumen *fax* berkurang karena dokumen *fax* yang dianggap penting saja yang dicetak oleh pengguna, seperti dokumen yang harus ditandatangani atau merupakan dokumen penting yang harus dikirim dalam bentuk *hardcopy*.

Tabel 4. Hasil Implementasi Penggunaan Aplikasi *Digifax*

Minggu Ke -	Sebelum		Setelah	
	Jumlah	Persentase	Jumlah	Persentase
1	8	100%	3	37,5%
2	12	100%	2	16,6%
3	7	100%	1	14,2%
4	10	100%	2	20%

Dari data tabel diatas terlihat pada setiap minggu selama satu bulan penggunaan kertas dalam penerimaan dokumen *fax* dengan menggunakan mesin *fax* sebanyak 100%. Setelah menggunakan aplikasi *digifax* penggunaan kertas berkurang perminggunya, meski seharusnya setelah penggunaan aplikasi *digifax* sebanyak 0% namun pada kenyataannya ada beberapa memang dokumen *fax* yang harus dicetak (*print*) seperti dokumen yang

harus ditandatangani maupun dokumen yang harus dikirim dalam bentuk *hardcopy*.

Dalam hal ini penulis melakukan perhitungan uji normalitas terlebih dahulu untuk menguji data yang didapat sebelum dan sesudah penggunaan kertas dengan uji *kolmogorov smirnov* dengan taraf signifikan 0,05 atau $\alpha = 5\%$ menggunakan aplikasi IBM SPSS Statistic Versi 21 didapat distribusi normal sebagai berikut:

	Sebelum	Sesudah
N	4	4
Normal Parameters ^{a,b}		
Mean	9,2500	2,0000
Std. Deviation	2,21736	,81650
Most Extreme Differences		
Absolute	,214	,250
Positive	,214	,250
Negative	-,155	-,250
Kolmogorov-Smirnov Z	,427	,500
Asymp. Sig. (2-tailed)	,993	,964

a. Test distribution is Normal.

b. Calculated from data.

Gambar 16. Hasil Uji Normalitas data dengan *Kolmogorov Smirnov*

5. KESIMPULAN

Berdasarkan tahap-tahap penelitian yang telah dilakukan aplikasi *Digifax* dapat menghemat penggunaan kertas, yang semula membutuhkan media kertas dan tinta untuk mencetak dokumen atau *file fax* yang diterima sebanyak 100% perminggunya namun setelah menggunakan aplikasi *digifax* dapat menghemat kertas sebanyak 37,5% pada minggu pertama, 16,6% pada minggu kedua, 14,2% pada minggu ketiga dan 20% pada minggu keempat. Hal ini karena dengan aplikasi *digifax* dokumen atau *file fax* yang diterima dapat disimpan dan dapat dicetak (*print*) sesuai kebutuhan pengguna.

Selain itu aplikasi *digifax* dapat menyimpan dokumen atau *file fax* yang sudah diterima dalam bentuk digital sebagai arsip atau dokumen penting bagi pengguna. Hal penting lainnya yaitu aplikasi *digifax* mempunyai sistem optimasi sebagai pilihan solusi yang tepat bagi pengguna, jika suatu saat tempat penyimpanan dokumen *fax* penuh maka dokumen *fax* dapat disimpan dalam *folder* arsip. Kemudian selain itu aplikasi *digifax* dapat mengelola dan memberikan hak akses terhadap arsip dokumen *fax* hanya untuk yang berkepentingan berdasarkan *user privilege* yang dapat dilakukan oleh seorang admin.

DAFTAR PUSTAKA

- SIMARMATA, J., 2000. Pengenalan Teknologi Komputer dan Informasi, Yogyakarta: Andi.
- SUTARMAN, S. M., 2009. Pengantar Teknologi Informasi, Jakarta: Bumi Aksara.
- INTERNET., 2013. <<http://www.engineeringtown.com/kids/index.php/penemuan/172-sejarah-ditemukannya-mesin-fax.Sabtu>> [Diakses 27 April 2013]

- KAMUS BESAR BAHASA INDONESIA, 1994. Jakarta . Balai Pustaka. hlm. 800
- MUNIR RENALDY, M., 2004. Algoritma Greedy. Bandung. Informatika.
- MALIK A., SHARMAA., & SAROHA V., 2013. Greedy Algorithm. International Journal of Scientific and Research Publications, ISSN: 250-3153, vol.3, no.8, 1-5.
- BECK K., 2002. Extreme Programming Explained: Embarce Change Ed.2. Wesley: Addison.
- PRESSMAN R. S., 2010. Rekayasa Perangkat Lunak terjemahan dari "Software Engineering" Ed.7. Yogyakarta: Andi.
- AMBARWARI A, YANTO W.N., 2016. Penerapan Algoritma Greedy pada Permasalahan Knapsack untuk Optimasi Pengangkutan Peti Kemas. Tersedia melalui : Research Gate
<https://www.researchgate.net/publication/290797775_Penerapan_Algoritma_Greedy_Pada_Permasalahan_Knapsack_Untuk_Optimasi_Pengangkutan_Peti_Kemas>
[Diakses 30 Juli 2018]
- HERLY M.A, RAHARJANA K.I, & PURBANDINI., 2015. Sistem Pencarian Hotel Berdasarkan Rute Perjalanan Terpendek dengan Mempertimbangkan Daya Tarik Wisata Menggunakan Algoritma Greedy. Journal of Information Systems Engineering and Business Intelligence. E-ISSN: 2443-2555, vol.1, no.1, 9-15.
- MANOR S., MEIR S., & ZHANG T., 2003. Greedy Algorithms for Classification-Consistency, Convergence-Rates, and Adaptivity. Journal of Machine Learning Research 4. 713-742.
- WAHYUNINGSIH P., 2018. Sistem Pencarian Hotel Berdasarkan Rute Perjalanan Terpendek dengan Mempertimbangkan Daya Tarik Wisata Menggunakan Algoritma Greedy. Journal of Information Systems En Penerapan Algoritma Greedy Untuk Mendeteksi Aktivitas Lansia Pada Karpet Menggunakan Arduino Mega. Jurnal INSTEK (Informatika Sains dan Teknologi). ISSN: 2541-1179, E-ISSN : 2581-1711, vol.3, no.1, 51-60.

Halaman ini sengaja dikosongkan