

PENGEMBANGAN SISTEM PEMBUKUAN DAN ADMINISTRASI PROTOKOL NOTARIS

(Studi Kasus: Kantor Notaris Dewi Andriani, S.H., M.H.)

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh
gelar Sarjana Komputer

Disusun oleh:

Krishna Yudhakuputra Putra Munandar

NIM: 155150207111004



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

**PENGEMBANGAN SISTEM PEMBUKUAN DAN ADMINISTRASI PROTOKOL NOTARIS
(STUDI KASUS: KANTOR NOTARIS DEWI ANDRIANI, S.H., M.H.)**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

**Disusun Oleh:
Krishna Yudhakusuma Putra Munandar
NIM: 155150207111004**

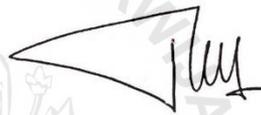
**Skripsi ini telah diuji dan dinyatakan lulus pada
29 Mei 2019
Telah diperiksa dan disetujui oleh:**

Dosen Pembimbing I



**Fajar Pradana, S.ST, M.Eng
NIP. 19871121 201504 1 004**

Dosen Pembimbing II



**Adam Hendra Brata, S.Kom, M.T,M.Sc
NIK. 2016079 00105 1 001**

**Mengetahui
Ketua Jurusan Teknik Informatika**



**Ti Astora Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001**



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Jakarta, 8 Januari 2019



Krishna Yudhakusuma P.M

NIM: 155150207111004



KATA PENGANTAR

Assalamu'alaikum Wr. Wb. Puji dan syukur penulis panjatkan atas rahmat dan karunia Allah SWT, penulis panjatkan kehadiran Allah SWT, karena atas berkah, rahmat, serta hidayah-Nya penulis dapat menyelesaikan menyelesaikan skripsi ini yang berjudul "Pengembangan Sistem Pembukuan dan Administrasi Protokol Notaris (Studi Kasus: Kantor Notaris Dewi Andriani, S.H., M.H.)". Skripsi ini diajukan sebagai syarat untuk mendapatkan gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya.

Selama proses penyusunan skripsi, penulis banyak belajar dan mengaplikasikan ilmu yang telah diperoleh dari perkuliahan. Semoga penulis dapat terus mengembangkan ilmu yang diperoleh selama perkuliahan dengan baik.

Pada kesempatan ini, penulis menyampaikan rasa terima kasih kepada pihak-pihak yang telah membantu penulis selama proses penyusunan skripsi, diantaranya:

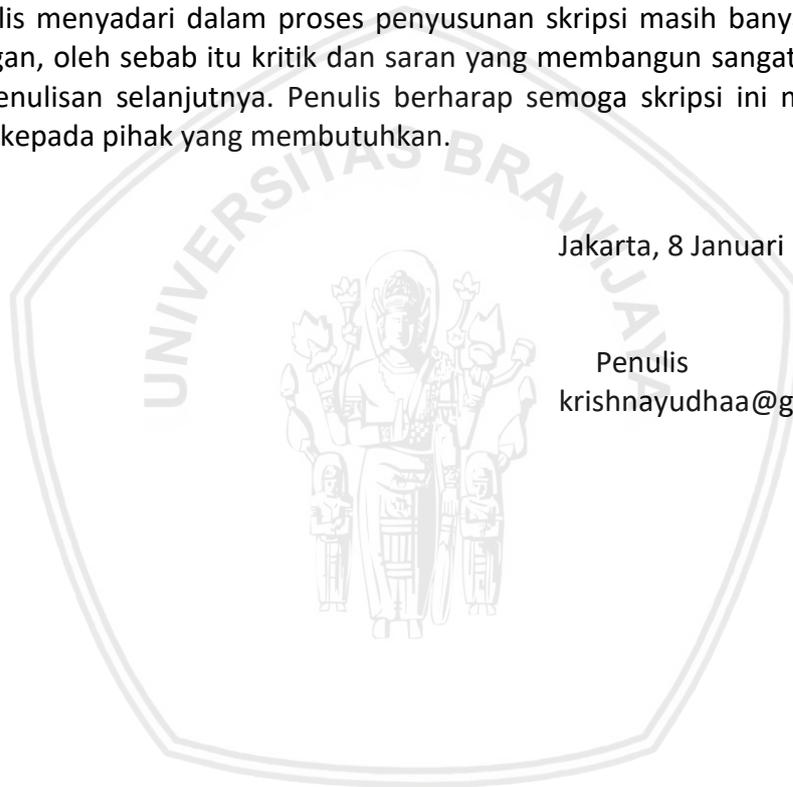
1. Allah SWT yang telah memberikan saya jalan yang terbaik dalam segala prosesnya.
2. Kepada kedua orang tua, adik, dan keluarga besar dari orang tua saya yang selalu mendukung saya baik dengan doa dan bantuan lainnya.
3. Bapak Fajar Pradana, S.ST, M.Eng selaku Pembimbing I, yang telah membimbing dan mengarahkan penulis dalam proses penyelesaian skripsi
4. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc. selaku Pembimbing II, yang telah membimbing dan mengarahkan penulis sehingga skripsi ini telah selesai.
5. Bapak dan Ibu dosen serta staff Fakultas Ilmu Komputer Universitas Brawijaya yang telah bersedia memberikan ilmu selama saya mengikuti perkuliahan.
6. Rekan-rekan Indonesian Future Leaders Chapter Malang yang telah memberikan wadah untuk mengembangkan *softskill* saya.
7. Arsyia Monica Pravina yang memberikan semangat, membantu, serta menemani dalam proses pengerjaan skripsi ini.
8. Kivlan Farizi, Ridwan Ganie, dan Mohammad Farhan yang selalu memberikan saran dalam pengerjaan skripsi ini, terutama dalam penulisan dan konsep sistem.
9. Teman-teman dekat saya yang selalu membantu saya dalam mendukung penuh dan memberikan saran dalam langkah-langkah saya mengerjakan skripsi.
10. Lucky Hidayat, Meylisa Dwi Marali, dan Nadya Ramadana yang selalu bersedia untuk menjawab pertanyaan saya terkait dengan pengerjaan skripsi.

11. Rekan-rekan Rumah Dakwah yang selalu memberikan semangat dalam mengerjakan skripsi ini.
12. Rekan-rekan EO-BIOS Fakultas Ilmu Komputer yang telah memberikan wadah untuk mengembangkan *softskill* saya selama dua tahun pertama kuliah.
13. Keluarga besar KAP yang membantu saya dalam proses perkuliahan saya di Fakultas Ilmu Komputer.
14. Teman-teman Fakultas Ilmu Komputer yang telah membantu selama masa studi penulis.
15. Seluruh pihak yang tidak dapat disebutkan satu per satu dan telah membantu selama proses penyusunan skripsi ini.

Penulis menyadari dalam proses penyusunan skripsi masih banyak terdapat kekurangan, oleh sebab itu kritik dan saran yang membangun sangat diperlukan dalam penulisan selanjutnya. Penulis berharap semoga skripsi ini memberikan manfaat kepada pihak yang membutuhkan.

Jakarta, 8 Januari 2019

Penulis
krishnayudhaa@gmail.com



ABSTRAK

Krishna Yudhokusuma Putra Munandar, Pengembangan Sistem Pembukuan dan Administrasi Protokol Notaris.

Pembimbing: Fajar Pradana, S.ST, M.Eng dan Adam Hendra Brata, S.Kom., M.T., M.Sc.

Notaris merupakan pejabat umum yang berwenang untuk membuat akta otentik dan kewenangan lainnya sebagaimana yang dimaksud dalam Undang-Undang Jabatan Notaris (UUJN). Salah satu kegiatan notaris adalah melakukan proses administrasi dan melakukan pembukuan terhadap akta yang disimpan berbentuk protokol notaris. Dari proses administrasi dan pembukuan protokol notaris ditemukan beberapa permasalahan yang dialami oleh pemilik dan pegawai kantor notaris. Permasalahan pertama adalah apabila pemilik notaris tidak dapat menjaga protokol notaris dengan baik, seperti tercecer, hilang, atau rusak pemilik notaris dapat terkena sanksi dari pasal 16 ayat (11) UUJN 2/2014. Permasalahan kedua, yaitu banyaknya protokol notaris dapat membuat kesulitan bagi pemilik dan pegawai notaris apabila sewaktu-waktu klien meminta perkembangan aktanya. Pemilik dan pegawai harus memperkirakan dimana akta tersebut disimpan. Permasalahan ketiga adalah tenggat pekerjaan menjadi masalah dikarenakan pegawai kantor notaris terkadang lupa pekerjaan mana yang mendekati tenggat. Selain itu, protokol notaris yang disimpan selama dapat mengalami kerusakan bertahun-tahun dikarenakan umur kertas hanya dapat bertahan beberapa belas tahun. Berdasarkan permasalahan tersebut, maka dilakukan pengembangan sistem pembukuan dan administrasi protokol notaris yang dapat menyimpan protokol notaris secara digital yang disesuaikan katalog terhadap penyimpanan fisik di kantor notaris, memberikan notifikasi umur kertas berdasarkan lama penyimpanan protokol notaris, pengingat tenggat pekerjaan dan pengawasan pekerjaan kantor, seperti pekerjaan apa saja yang belum selesai, perkembangan dari pekerjaan yang sedang dikerjakan, dan siapa yang mengerjakan menyelesaikan permasalahan kantor notaris. Proses pengembangan perangkat lunak ini menggunakan metode *waterfall*. Sistem ini dikembangkan dengan berbasis web, menggunakan bahasa pemrograman PHP, dan *framework codeigniter*. Pengujian unit dilakukan dengan menggunakan metode *whitebox testing* dan hasilnya menunjukan nilai 100% valid dari 3 kasus uji. Pengujian Validasi dilakukan menggunakan metode *black box testing* dan hasilnya 100% valid dari 93 kasus uji. Pengujian *usability* dilakukan dengan metode *System Usability Scale* dan mendapatkan skor sebesar 85,5 dan dari skor ini dapat dikategorikan *acceptable*, sehingga sistem telah memenuhi kebutuhan pengguna.

Kata Kunci: *Notaris, pengembangan perangkat lunak, Waterfall, PHP, Codeigniter, White box testing, Black box testing, Usability testing*

ABSTRACT

Krishna Yudhakuputra Munandar, Pengembangan Sistem Pembukuan dan Administrasi Protokol Notaris.

Adviser: Fajar Pradana, S.ST, M.Eng dan Adam Hendra Brata, S.Kom., M.T., M.Sc.

A notary is an approved public official to make an authentic deed and other authority, written in Undang-Undang Jabatan Notaris (UUJN). One of the notary activity is to carry out the administrative process and book the deeds that stored in the form of a notary protocol. From the administration process to bookkeeping notary protocols, there are several problems experienced by owners and employees of the notary office. The first problem is when the notary owners cannot properly maintain the notary protocol, such as being scattered, lost, or damaged the notary may be sanctioned is written in pasal 16 article number (11) UUJN 2/2014. The second problem is the number of notary protocols, it is difficult for owners and employees when the client ask their deed progress. The Owners and Employees must predict where the deed is stored. The third problem is deadlines, because employees of the notary office sometimes forget which work is close to the deadline. In addition, notary protocols that are stored can be damaged for years due to the age of the paper that can only last a dozen years. Based on these problems, there's a new method for the bookkeeping system and notary protocol administration, is stored digitally. Based on the notary protocols catalog from the physical storage at the office, it gives notifications about the paper age, deadline reminders and supervise the office work, such as any work that has not been completed, the progress of the work, and who works to solve the office problems. The software development is using the waterfall method. This system was developed by web-based, using the PHP programming language, and codeigniter framework. Unit testing is using the white box testing method and the results 100% valid from 3 tests. Validation testing is done using the black box testing method and the results are 100% valid from 93 tests. Usability testing is done using the System Usability Scale method the score is 85,5 and categorized acceptable, so the system has complete the needs of users.

Keyword: Notary, Software Development, Waterfall, PHP, Codeigniter, White box testing, Black box testing, Usability testing.

DAFTAR ISI

PENGEMBANGAN SISTEM PEMBUKUAN DAN ADMINISTRASI PROTOKOL NOTARIS	1
(Studi Kasus: Kantor Notaris Dewi Andriani, S.H., M.H.)	1
PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR.....	iii
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xxii
DAFTAR LAMPIRAN	xxiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Rekayasa Perangkat Lunak	6
2.3 Pengembangan Perangkat Lunak	6
2.3.1 Metode <i>Waterfall</i>	6
2.3.2 Pendekatan Berorientasi Objek	8
2.3.3 UML (<i>Unified Modelling Language</i>)	8
2.4 Sistem Pembukuan Protokol Notaris.....	10
2.5 Sistem Administrasi Protokol Notaris	10
2.6 Profesi Notaris	10
2.7 Protokol Notaris.....	11
2.8 Model-view-controller	11



2.9 Codeigniter.....	12
2.10 Pengujian Perangkat Lunak	12
2.10.1 Pengujian Unit.....	12
2.10.2 Pengujian Integrasi.....	13
2.10.3 Pengujian Validasi	13
2.10.4 Pengujian <i>Usability</i>	14
BAB 3 METODOLOGI PENELITIAN	16
3.1 Studi Literatur	16
3.2 Analisis Kebutuhan	17
3.3 Perancangan Sistem.....	17
3.4 Implementasi	17
3.5 Pengujian	17
3.6 Kesimpulan dan Saran	18
BAB 4 ANALISIS KEBUTUHAN	19
4.1 Deskripsi Sistem	19
4.2 Identifikasi Aktor.....	19
4.3 Kebutuhan Fungsional	20
4.4 Kebutuhan Non-Fungsional	25
4.4.1 Diagram <i>Use Case</i>	25
4.5 <i>Use Case Scenario</i>	27
BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM	65
5.1 Perancangan Sistem.....	65
5.1.1 Perancangan <i>Sequence Diagram</i>	66
5.1.2 Perancangan Class Diagram	68
5.1.3 <i>Perancangan Algoritme</i>	70
5.1.4 Perancangan Basis Data	72
5.1.5 Perancangan Antarmuka.....	77
5.2 Implementasi Sistem	88
5.2.1 Spesifikasi Pengembangan Sistem	88
5.2.2 Implementasi Basis Data	89
5.2.3 Implementasi Kode Program	90
5.2.4 Implementasi Antarmuka	93



BAB 6 PENGUJIAN	96
6.1 Pengujian Unit.....	96
6.1.1 Pengujian Unit Kelas Order pada Fungsi tambahOrder()	96
6.1.2 Pengujian Unit Kelas Helper pada Fungsi Work()	98
6.1.3 Pengujian Unit Kelas Helper pada Fungsi Paper().....	102
6.2 Pengujian Integrasi	105
6.3 Pengujian Validasi	110
6.3.1 Pengujian Validasi <i>Login</i>	110
6.3.2 Pengujian Validasi Logout	111
6.3.3 Pengujian Validasi Melihat Akun Pegawai	112
6.3.4 Pengujian Validasi Menambah Akun Pegawai	112
6.3.5 Pengujian Validasi Mengubah Akun Pegawai	113
6.3.6 Pengujian Validasi Menghapus Akun Pegawai	114
6.3.7 Pengujian Validasi Menambah Klien.....	115
6.3.8 Pengujian Validasi Melihat Klien.....	116
6.3.9 Pengujian Validasi Mengubah Klien.....	116
6.3.10 Pengujian Validasi Melihat Riwayat Klien	117
6.3.11 Pengujian Validasi Melihat Pesan Nama Organisasi	117
6.3.12 Pengujian Validasi Menambah Pesan Nama Organisasi.....	117
6.3.13 Pengujian Validasi Melihat Rincian Berkas Klien pada Pesan Nama Organisasi	118
6.3.14 Pengujian Validasi Mengubah Pesan Nama Organisasi	119
6.3.15 Pengujian Validasi Menolak Pesan Nama Organisasi	119
6.3.16 Pengujian Validasi Menghapus Pesan Nama Organisasi	120
6.3.17 Pengujian Validasi Memverifikasi Pesan Nama Organisasi	120
6.3.18 Pengujian Validasi Melihat Buat Akta Organisasi	121
6.3.19 Pengujian Validasi Melihat Rincian Berkas Klien pada Buat Akta Organisasi	121
6.3.20 Pengujian Validasi Membarui Status Buat Akta Organisasi	122
6.3.21 Pengujian Validasi Mengonfirmasi Status Buat Akta Organisasi Selesai.....	122
6.3.22 Pengujian Validasi Melihat Pendaftaran SK Organisasi	122
6.3.23 Pengujian Validasi Mengunggah Berkas SK Organisasi.....	123



6.3.24 Pengujian Validasi Melihat Rincian Berkas Klien pada Pendaftaran SK Organisasi	123
6.3.25 Pengujian Validasi Melihat Cek nama Wasiat.....	124
6.3.26 Pengujian Validasi Menambah Cek nama Wasiat.....	124
6.3.27 Pengujian Validasi Melihat Rincian Berkas Klien pada Cek nama Wasiat.....	125
6.3.28 Pengujian Validasi Memverifikasi Cek Nama Wasiat.....	126
6.3.29 Pengujian Validasi Menolak Cek Nama Wasiat.....	126
6.3.30 Pengujian Validasi Mengubah Cek nama Wasiat.....	127
6.3.31 Pengujian Validasi Memverifikasi Status Pewasiat.....	127
6.3.32 Pengujian Validasi Melihat Buat Akta Wasiat.....	128
6.3.33 Pengujian Validasi Melihat Rincian Berkas Klien pada Buat Akta Wasiat.....	128
6.3.34 Pengujian Validasi Membarui Status Buat Akta Wasiat	129
6.3.35 Pengujian Validasi Mengonfirmasi Status Buat Akta Wasiat Selesai.....	129
6.3.36 Pengujian Validasi Melihat Pendaftaran SK Wasiat.....	129
6.3.37 Pengujian Validasi Mengunggah Berkas SK Wasiat	130
6.3.38 Pengujian Validasi Melihat Rincian Berkas Klien pada Pendaftaran SK Wasiat.....	131
6.3.39 Pengujian Validasi Melihat Buat Akta Fidusia.....	131
6.3.40 Pengujian Validasi Menambah Buat Akta Fidusia.....	131
6.3.41 Pengujian Validasi Melihat Rincian Berkas Klien pada Buat akta Fidusia	132
6.3.42 Pengujian Validasi Memverifikasi Buat Akta Fidusia	133
6.3.43 Pengujian Validasi Menolak Buat Akta Fidusia	133
6.3.44 Pengujian Validasi Mengubah Buat Akta Fidusia.....	134
6.3.45 Pengujian Validasi Membarui Status Buat Akta Fidusia	135
6.3.46 Pengujian Validasi Mengonfirmasi Status Buat Akta Fidusia Selesai.....	135
6.3.47 Pengujian Validasi Melihat Pendaftaran SK Fidusia.....	135
6.3.48 Pengujian Validasi Mengunggah Berkas SK Fidusia	136
6.3.49 Pengujian Validasi Melihat Rincian Berkas Klien pada Pendaftaran SK Fidusia.....	136



6.3.50 Pengujian Validasi Melihat Rincian Berkas Klien pada Protokol Notaris.....	137
6.3.51 Pengujian Validasi Mengubah Protokol Notaris	137
6.3.52 Pengujian Validasi Menambah Protokol Notaris	138
6.3.53 Pengujian Validasi Melihat Protokol Notaris	139
6.3.54 Pengujian Validasi Melihat Dashboard	140
6.3.55 Pengujian Validasi Melihat Monitoring Kerja	140
6.3.56 Pengujian Validasi Melihat Log Aktivitas Pegawai.....	140
6.3.57 Pengujian Validasi Melihat Notifikasi <i>Deadline</i>	140
6.3.58 Pengujian Validasi Melihat Notifikasi Umur Kertas	141
6.3.59 Pengujian Validasi Menambah Informasi	142
6.3.60 Pengujian Validasi Mengubah Informasi	143
6.3.61 Pengujian Validasi Menghapus Informasi.....	143
6.4 Pengujian <i>Usability</i>	144
6.4.1 Prosedur Pengujian <i>Usability</i>	144
6.4.2 Analisis dan Hasil Pengujian <i>Usability</i>	145
BAB 7 PENUTUP	147
7.1 Kesimpulan.....	147
7.2 Saran	147
DAFTAR REFERENSI	148
LAMPIRAN A SURAT BUKTI ELISITASI KEBUTUHAN	150
LAMPIRAN B HASIL WAWANCARA.....	151
LAMPIRAN C KUESIONER <i>SYSTEM USABILITY SCALE</i>	154
LAMPIRAN D RESPONDEN <i>SYSTEM USABILITY SCALE</i>	156

DAFTAR TABEL

Tabel 2.1 Elemen <i>sequence diagram</i>	9
Tabel 2.2 Keterangan poin skala likert.....	14
Tabel 4.1 Identifikasi aktor sistem	19
Tabel 4.1 Identifikasi aktor sistem (lanjutan)	20
Tabel 4.2 Kebutuhan fungsional	20
Tabel 4.2 Kebutuhan fungsional (lanjutan).....	21
Tabel 4.2 Kebutuhan fungsional (lanjutan).....	22
Tabel 4.2 Kebutuhan fungsional (lanjutan).....	23
Tabel 4.2 Kebutuhan fungsional (lanjutan).....	24
Tabel 4.2 Kebutuhan fungsional (lanjutan).....	25
Tabel 4.3 Kebutuhan non-fungsional.....	25
Tabel 4.4 <i>Use case scenario</i> login.....	27
Tabel 4.5 <i>Use case scenario</i> logout.....	27
Tabel 4.5 <i>Use case scenario</i> logout (lanjutan)	28
Tabel 4.6 <i>Use case scenario</i> melihat akun pegawai.....	28
Tabel 4.7 <i>Use case scenario</i> menambah akun pegawai.....	28
Tabel 4.7 <i>Use case scenario</i> menambah akun pegawai (lanjutan).....	29
Tabel 4.8 <i>Use case scenario</i> mengubah akun pegawai.....	29
Tabel 4.8 <i>Use case scenario</i> mengubah akun pegawai (lanjutan).....	30
Tabel 4.9 <i>Use case scenario</i> menghapus akun pegawai	30
Tabel 4.9 <i>Use case scenario</i> menghapus akun (lanjutan).....	31
Tabel 4.10 <i>Use case scenario</i> menambah klien	31
Tabel 4.11 <i>Use case scenario</i> melihat klien	31
Tabel 4.11 <i>Use case scenario</i> melihat klien (lanjutan).....	32
Tabel 4.12 <i>Use case scenario</i> mengubah klien	32
Tabel 4.13 <i>Use case scenario</i> melihat riwayat klien	33
Tabel 4.14 <i>Use case scenario</i> melihat pesan nama organisasi	33
Tabel 4.15 <i>Use case scenario</i> menambah pesan nama organisasi	34
Tabel 4.16 <i>Use case scenario</i> melihat rincian berkas klien pada pesan nama organisasi	34

Tabel 4.16 <i>Use case scenario</i> melihat rincian berkas klien pada pesan nama organisasi (lanjutan).....	35
Tabel 4.17 <i>Use case scenario</i> mengubah pesan nama organisasi	35
Tabel 4.18 <i>Use case scenario</i> menolak pesan nama organisasi.....	36
Tabel 4.19 <i>Use case scenario</i> menghapus pesan nama organisasi.....	36
Tabel 4.19 <i>Use case scenario</i> menghapus pesan nama organisasi (lanjutan)	37
Tabel 4.20 <i>Use case scenario</i> memverifikasi pesan nama organisasi.....	37
Tabel 4.21 <i>Use case scenario</i> melihat buat akta organisasi	37
Tabel 4.21 <i>Use case scenario</i> melihat buat akta organisasi (lanjutan).....	38
Tabel 4.22 <i>Use case scenario</i> melihat rincian berkas klien pada buat akta organisasi	38
Tabel 4.23 <i>Use case scenario</i> membarui status buat akta organisasi	38
Tabel 4.23 <i>Use case scenario</i> membarui status buat akta organisasi (lanjutan)..	39
Tabel 4.24 <i>Use case scenario</i> mengonfirmasi status buat akta organisasi selesai	39
Tabel 4.24 <i>Use case scenario</i> mengonfirmasi status buat akta organisasi selesai (lanjutan).....	40
Tabel 4.25 <i>Use case scenario</i> melihat pendaftaran SK organisasi	40
Tabel 4.26 <i>Use case scenario</i> mengunggah berkas SK organisasi.....	40
Tabel 4.26 <i>Use case scenario</i> mengunggah berkas SK organisasi (lanjutan).....	41
Tabel 4.27 <i>Use case scenario</i> melihat rincian berkas klien pada pendaftaran SK organisasi	41
Tabel 4.27 <i>Use case scenario</i> melihat rincian berkas klien pada pendaftaran SK organisasi (lanjutan).....	42
Tabel 4.28 <i>Use case scenario</i> melihat cek nama wasiat	42
Tabel 4.29 <i>Use case scenario</i> menambah cek nama wasiat	42
Tabel 4.29 <i>Use case scenario</i> menambah cek nama wasiat (lanjutan)	43
Tabel 4.30 <i>Use case scenario</i> melihat rincian berkas klien pada cek nama wasiat	43
Tabel 4.31 <i>Use case scenario</i> memverifikasi cek nama wasiat.....	44
Tabel 4.32 <i>Use case scenario</i> menolak cek nama wasiat.....	44
Tabel 4.32 <i>Use case scenario</i> menolak cek nama wasiat (lanjutan).....	45
Tabel 4.33 <i>Use case scenario</i> mengubah cek nama wasiat	45
Tabel 4.34 <i>Use case scenario</i> memverifikasi status pewasiat	45
Tabel 4.34 <i>Use case scenario</i> memverifikasi status pewasiat (lanjutan).....	46

Tabel 4.35 <i>Use case scenario</i> melihat buat akta wasiat	46
Tabel 4.36 <i>Use case scenario</i> melihat rincian berkas klien pada buat akta wasiat	47
Tabel 4.37 <i>Use case scenario</i> membarui status buat akta wasiat	47
Tabel 4.38 <i>Use case scenario</i> mengonfirmasi status buat akta wasiat selesai.....	48
Tabel 4.39 <i>Use case scenario</i> melihat pendaftaran SK wasiat.....	48
Tabel 4.39 <i>Use case scenario</i> melihat pendaftaran SK wasiat (lanjutan)	49
Tabel 4.40 <i>Use case scenario</i> mengunggah berkas SK wasiat	49
Tabel 4.41 <i>Use case scenario</i> melihat rincian berkas klien pada pendaftaran SK wasiat	49
Tabel 4.41 <i>Use case scenario</i> melihat rincian berkas klien pada pendaftaran SK wasiat (lanjutan).....	50
Tabel 4.42 <i>Use case scenario</i> melihat buat akta fidusia	50
Tabel 4.42 <i>Use case scenario</i> melihat buat akta fidusia (lanjutan)	51
Tabel 4.43 <i>Use case scenario</i> menambah buat akta fidusia	51
Tabel 4.44 <i>Use case scenario</i> melihat rincian berkas klien pada buat akta fidusia	52
Tabel 4.45 <i>Use case scenario</i> memverifikasi buat akta fidusia.....	52
Tabel 4.46 <i>Use case scenario</i> menolak buat akta fidusia	53
Tabel 4.47 <i>Use case scenario</i> mengubah buat akta fidusia	53
Tabel 4.47 <i>Use case scenario</i> mengubah buat akta fidusia (lanjutan)	54
Tabel 4.48 <i>Use case scenario</i> membarui status buat akta fidusia	54
Tabel 4.49 <i>Use case scenario</i> mengonfirmasi status buat akta fidusia selesai	54
Tabel 4.49 <i>Use case scenario</i> mengonfirmasi status buat akta fidusia selesai (lanjutan).....	55
Tabel 4.50 <i>Use case scenario</i> melihat pendaftaran SK fidusia.....	55
Tabel 4.51 <i>Use case scenario</i> mengunggah berkas SK fidusia	56
Tabel 4.52 <i>Use case scenario</i> melihat rincian berkas klien pada pendaftaran SK fidusia	56
Tabel 4.52 <i>Use case scenario</i> melihat rincian berkas klien pada pendaftaran SK fidusia (lanjutan)	57
Tabel 4.53 <i>Use case scenario</i> melihat rincian berkas klien pada protokol notaris	57
Tabel 4.54 <i>Use case scenario</i> mengubah protokol notaris	57
Tabel 4.54 <i>Use case scenario</i> mengubah protokol notaris (lanjutan)	58



Tabel 4.55 <i>Use case scenario</i> menambah protokol notaris.....	58
Tabel 4.55 <i>Use case scenario</i> menambah protokol notaris (lanjutan)	59
Tabel 4.56 <i>Use case scenario</i> melihat protokol notaris	59
Tabel 4.57 <i>Use case scenario</i> melihat dashboard	59
Tabel 4.57 <i>Use case scenario</i> melihat dashboard (lanjutan)	60
Tabel 4.58 <i>Use case scenario</i> melihat monitoring kerja	60
Tabel 4.59 <i>Use case scenario</i> melihat log aktivitas pegawai	60
Tabel 4.59 <i>Use case scenario</i> melihat log aktivitas pegawai (lanjutan).....	61
Tabel 4.60 <i>Use case scenario</i> melihat notifikasi deadline	61
Tabel 4.61 <i>Use case scenario</i> melihat notifikasi umur kertas.....	61
Tabel 4.61 <i>Use case scenario</i> melihat notifikasi umur kertas (lanjutan)	62
Tabel 4.62 <i>Use case scenario</i> menambah informasi.....	62
Tabel 4.63 <i>Use case scenario</i> mengubah informasi.....	62
Tabel 4.63 <i>Use case scenario</i> mengubah informasi (lanjutan)	63
Tabel 4.64 <i>Use case scenario</i> menghapus informasi	63
Tabel 4.64 <i>Use case scenario</i> menghapus informasi (lanjutan).....	64
Tabel 5.1 Algoritme fungsi tambahOrder	71
Tabel 5.2 Algoritme fungsi work	71
Tabel 5.2 Algoritme fungsi work (lanjutan)	72
Tabel 5.3 Algoritme fungsi paper	72
Tabel 5.4 Struktur tabel informasi	73
Tabel 5.5 Struktur tabel akun.....	74
Tabel 5.6 Struktur tabel log_aktivitas_pegawai.....	74
Tabel 5.7 Struktur tabel klien	74
Tabel 5.7 Struktur tabel klien (lanjutan)	75
Tabel 5.8 Struktur tabel orderjob	75
Tabel 5.9 Struktur tabel instansi	75
Tabel 5.10 Struktur tabel orders	75
Tabel 5.10 Struktur tabel orders (lanjutan)	76
Tabel 5.11 Struktur tabel protokol.....	76
Tabel 5.12 Penjelasan perancangan antarmuka halaman menambah order organisasi	77



Tabel 5.12 Penjelasan perancangan antarmuka halaman menambah order (lanjutan).....	78
Tabel 5.12 Penjelasan perancangan antarmuka halaman menambah order (lanjutan).....	79
Tabel 5.13 Penjelasan perancangan antarmuka halaman login	79
Tabel 5.13 Penjelasan perancangan antarmuka halaman login (lanjutan)	80
Tabel 5.14 Penjelasan perancangan antarmuka halaman melihat monitoring kerja	80
Tabel 5.14 Penjelasan perancangan antarmuka halaman melihat monitoring kerja (lanjutan).....	81
Tabel 5.14 Penjelasan perancangan antarmuka halaman melihat monitoring kerja (lanjutan).....	82
Tabel 5.15 Penjelasan perancangan antarmuka halaman melihat klien	82
Tabel 5.15 Penjelasan perancangan antarmuka halaman melihat klien (lanjutan)	83
Tabel 5.15 Penjelasan perancangan antarmuka halaman melihat klien (lanjutan)	84
Tabel 5.16 Penjelasan perancangan antarmuka halaman melihat akun.....	85
Tabel 5.16 Penjelasan perancangan antarmuka halaman melihat akun (lanjutan)	86
Tabel 5.17 Penjelasan perancangan antarmuka halaman membuat akun	87
Tabel 5.17 Penjelasan perancangan antarmuka halaman membuat akun (lanjutan)	88
Tabel 5.18 Spesifikasi perangkat keras	89
Tabel 5.19 Spesifikasi perangkat lunak	89
Tabel 5.20 Kode program fungsi tambahOrder	90
Tabel 5.20 Kode program fungsi tambahOrder (lanjutan)	91
Tabel 5.21 Kode program fungsi work	91
Tabel 5.21 Kode program fungsi work (lanjutan)	92
Tabel 5.22 Kode program fungsi paper.....	92
Tabel 5.22 Kode program fungsi paper (lanjutan)	93
Tabel 6.1 Pseudocode tambahOrder	96
Tabel 6.1 Pseudocode tambahOrder (lanjutan)	97
Tabel 6.2 Hasil pengujian unit fungsi tambahOrder() kelas Order	98
Tabel 6.3 Pseudocode work	98



Tabel 6.3 Pseudocode work (lanjutan)	99
Tabel 6.4 Hasil pengujian unit fungsi work() kelas Helper	100
Tabel 6.4 Hasil pengujian unit fungsi work() kelas Helper (lanjutan)	101
Tabel 6.4 Hasil pengujian unit fungsi work() kelas Helper (lanjutan)	102
Tabel 6.5 Pseudocode paper	102
Tabel 6.6 Hasil pengujian unit fungsi paper() kelas Helper	103
Tabel 6.6 Hasil pengujian unit fungsi paper() kelas Helper (lanjutan).....	104
Tabel 6.6 Hasil pengujian unit fungsi paper() kelas Helper (lanjutan).....	105
Tabel 6.7 Pseudocode tambahOrder	105
Tabel 6.7 Pseudocode tambahOrder (lanjutan)	106
Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order.....	107
Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order (lanjutan)	108
Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order (lanjutan)	109
Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order (lanjutan)	110
Tabel 6.9 Kasus uji berhasil melakukan login.....	110
Tabel 6.9 Kasus uji berhasil melakukan login (lanjutan)	111
Tabel 6.10 Kasus uji gagal melakukan login dengan terdapat kekosongan pada data yang dibutuhkan sistem	111
Tabel 6.11 Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom username atau password	111
Tabel 6.12 Kasus uji berhasil logout.....	111
Tabel 6.12 Kasus uji berhasil logout (lanjutan)	112
Tabel 6.13 Kasus uji berhasil melihat akun pegawai	112
Tabel 6.14 Kasus uji berhasil menambah akun pegawai	112
Tabel 6.15 Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem	112
Tabel 6.15 Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem (lanjutan)	113
Tabel 6.16 Kasus uji gagal menambah akun dengan username yang sudah terdapat pada daftar akun	113
Tabel 6.17 Kasus uji berhasil mengubah akun pegawai	113

Tabel 6.18 Kasus uji gagal mengubah akun pegawai dengan terdapat kekosongan pada data yang dibutuhkan sistem	114
Tabel 6.19 Kasus uji berhasil menghapus akun pegawai	114
Tabel 6.20 Kasus uji gagal menghapus akun pegawai	114
Tabel 6.21 Kasus uji berhasil menambah klien	115
Tabel 6.22 Kasus uji gagal menambah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem	115
Tabel 6.23 Kasus uji gagal menambah klien dengan nomor KTP yang sudah terdapat pada daftar klien	115
Tabel 6.24 Kasus uji berhasil melihat klien	116
Tabel 6.25 Kasus uji berhasil mengubah klien	116
Tabel 6.26 Kasus uji gagal mengubah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem	116
Tabel 6.27 Kasus uji berhasil melihat riwayat klien	117
Tabel 6.28 Kasus uji berhasil melihat pesan nama organisasi	117
Tabel 6.29 Kasus uji berhasil menambah pesan nama organisasi	117
Tabel 6.30 Kasus uji gagal menambah pesan nama organisasi dengan file yang diunggah bukan ekstensi .pdf	118
Tabel 6.31 Kasus uji gagal menambah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem	118
Tabel 6.32 Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi	118
Tabel 6.32 Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi (lanjutan)	119
Tabel 6.33 Kasus uji berhasil mengubah pesan nama organisasi	119
Tabel 6.34 Kasus uji gagal mengubah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem	119
Tabel 6.35 Kasus uji berhasil menolak pesan nama organisasi	119
Tabel 6.35 Kasus uji berhasil menolak pesan nama organisasi (lanjutan)	120
Tabel 6.36 Kasus uji gagal menolak pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem	120
Tabel 6.37 Kasus uji berhasil menghapus pesan nama organisasi	120
Tabel 6.38 Kasus uji berhasil memverifikasi pesan nama organisasi	121
Tabel 6.39 Kasus uji berhasil melihat buat akta organisasi	121
Tabel 6.40 Kasus uji berhasil melihat rincian berkas klien pada buat akta organisasi	121

Tabel 6.41 Kasus uji berhasil membarui status buat akta organisasi	122
Tabel 6.42 Kasus uji berhasil membarui status buat akta organisasi selesai	122
Tabel 6.43 Kasus uji berhasil melihat pendaftaran SK organisasi.....	122
Tabel 6.44 Kasus uji berhasil mengunggah berkas SK organisasi	123
Tabel 6.45 Kasus uji berhasil mengunggah berkas SK organisasi dengan file yang diunggah bukan ekstensi .pdf.	123
Tabel 6.46 Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK organisasi	124
Tabel 6.47 Kasus uji berhasil melihat cek nama wasiat	124
Tabel 6.48 Kasus uji berhasil menambah cek nama wasiat.....	124
Tabel 6.49 Kasus uji gagal menambah cek nama wasiat dengan file yang diunggah bukan ekstensi .pdf	125
Tabel 6.50 Kasus uji gagal menambah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.....	125
Tabel 6.51 Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat	125
Tabel 6.51 Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat (lanjutan).....	126
Tabel 6.52 Kasus uji berhasil memverifikasi cek nama wasiat	126
Tabel 6.53 Kasus uji berhasil menolak cek nama wasiat	126
Tabel 6.54 Kasus uji gagal menolak cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem	126
Tabel 6.54 Kasus uji gagal menolak cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem (lanjutan)	127
Tabel 6.55 Kasus uji berhasil mengubah cek nama wasiat	127
Tabel 6.56 Kasus uji gagal mengubah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.....	127
Tabel 6.57 Kasus uji berhasil memverifikasi status pewasiat	128
Tabel 6.58 Kasus uji berhasil melihat buat akta wasiat	128
Tabel 6.59 Kasus uji berhasil melihat rincian berkas klien pada buat akta wasiat	128
Tabel 6.60 Kasus uji berhasil membarui status buat akta wasiat	129
Tabel 6.61 Kasus uji berhasil mengonfirmasi status buat akta wasiat selesai ...	129
Tabel 6.62 Kasus uji berhasil melihat pendaftaran SK wasiat.....	129
Tabel 6.63 Kasus uji berhasil mengunggah berkas SK wasiat	130



Tabel 6.64 Kasus uji berhasil mengunggah berkas SK wasiat dengan file yang diunggah bukan ekstensi .pdf	130
Tabel 6.65 Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK wasiat	131
Tabel 6.66 Kasus uji berhasil melihat buat akta fidusia	131
Tabel 6.67 Kasus uji berhasil menambah buat akta fidusia.....	131
Tabel 6.67 Kasus uji berhasil menambah buat akta fidusia (lanjutan)	132
Tabel 6.68 Kasus uji gagal menambah buat akta fidusia dengan file yang diunggah bukan ekstensi .pdf	132
Tabel 6.69 Kasus uji gagal menambah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.....	132
Tabel 6.70 Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia	132
Tabel 6.70 Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia (lanjutan)	133
Tabel 6.71 Kasus uji berhasil memverifikasi buat akta fidusia	133
Tabel 6.72 Kasus uji berhasil menolak buat akta fidusia	133
Tabel 6.73 Kasus uji gagal menolak buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem	134
Tabel 6.74 Kasus uji berhasil mengubah buat akta fidusia	134
Tabel 6.75 Kasus uji gagal mengubah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.....	134
Tabel 6.76 Kasus uji berhasil membarui status buat akta fidusia.....	135
Tabel 6.77 Kasus uji berhasil membarui status buat akta fidusia selesai	135
Tabel 6.78 Kasus uji berhasil melihat pendaftaran SK fidusia	135
Tabel 6.79 Kasus uji berhasil mengunggah berkas SK fidusia	136
Tabel 6.80 Kasus uji berhasil mengunggah berkas SK fidusia dengan file yang diunggah bukan ekstensi .pdf	136
Tabel 6.81 Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK fidusia	137
Tabel 6.82 Kasus uji berhasil melihat rincian berkas klien pada protokol notaris	137
Tabel 6.83 Kasus uji berhasil mengubah protokol notaris.....	137
Tabel 6.83 Kasus uji berhasil mengubah protokol notaris (lanjutan)	138
Tabel 6.84 Kasus uji gagal mengubah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.....	138

Tabel 6.85 Kasus uji berhasil menambah protokol notaris.....	138
Tabel 6.86 Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.....	138
Tabel 6.86 Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem (lanjutan).....	139
Tabel 6.87 Kasus uji gagal menambah protokol notaris dengan nomor protokol yang sudah terdapat pada daftar protokol notaris.....	139
Tabel 6.88 Kasus uji berhasil melihat protokol notaris.....	139
Tabel 6.89 Kasus uji berhasil melihat <i>dashboard</i>	140
Tabel 6.90 Kasus uji berhasil melihat monitoring kerja.....	140
Tabel 6.91 Kasus uji berhasil melihat log aktivitas pegawai	140
Tabel 6.92 Kasus uji berhasil melihat notifikasi <i>deadline</i>	140
Tabel 6.92 Kasus uji berhasil melihat notifikasi <i>deadline</i> (lanjutan).....	141
Tabel 6.93 Kasus uji berhasil melihat notifikasi <i>deadline</i> dengan tidak ada notifikasi <i>deadline</i>	141
Tabel 6.94 Kasus uji berhasil melihat notifikasi umur kertas	141
Tabel 6.95 Kasus uji berhasil melihat notifikasi umur kertas dengan tidak ada notifikasi umur kertas	142
Tabel 6.96 Kasus uji berhasil menambah informasi	142
Tabel 6.97 Kasus uji gagal menambah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem	142
Tabel 6.98 Kasus uji berhasil mengubah informasi.....	143
Tabel 6.99 Kasus uji gagal mengubah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem	143
Tabel 6.100 Kasus uji berhasil menghapus informasi	143
Tabel 6.101 Kasus uji gagal menghapus informasi	144
Tabel 6.102 Daftar Pertanyaan SUS	144
Tabel 6.102 Daftar Pertanyaan SUS (lanjutan)	145
Tabel 6.103 Hasil kuesioner skor tiap pertanyaan SUS.....	145
Tabel 6.104 Hasil kuesioner skor tiap pertanyaan SUS.....	146

DAFTAR GAMBAR

Gambar 2.1 Model <i>waterfall</i>	7
Gambar 2.2 <i>Use case diagram</i>	9
Gambar 2.3 Rating dan skala konversi poin rata-rata SUS	15
Gambar 3.1 Diagram alir penelitian	16
Gambar 4.1 Diagram <i>use case</i> sistem	26
Gambar 5.1 Perancangan diagram alir sistem	65
Gambar 5.2 <i>Sequence diagram</i> menambah pesan nama organisasi.....	66
Gambar 5.3 <i>Sequence diagram</i> melihat notifikasi <i>deadline</i>	67
Gambar 5.4 <i>Sequence diagram</i> melihat notifikasi umur kertas	68
Gambar 5.5 <i>Class diagram</i> perancangan umum.....	69
Gambar 5.6 <i>Class diagram</i> pada Model.....	69
Gambar 5.7 <i>Class diagram</i> pada Controller.....	70
Gambar 5.8 <i>Entity Relationship Diagram</i> (ERD).....	73
Gambar 5.9 Perancangan antarmuka halaman menambah order organisasi.....	77
Gambar 5.10 Perancangan antarmuka halaman login	79
Gambar 5.11 Perancangan antarmuka halaman melihat monitoring kerja	80
Gambar 5.12 Perancangan antarmuka halaman melihat klien	82
Gambar 5.13 Perancangan antarmuka halaman melihat akun	84
Gambar 5.14 Perancangan antarmuka halaman membuat akun.....	86
Gambar 5.15 Skema basis data sistem pembukuan dan administrasi protokol notaris	90
Gambar 5.16 Implementasi antarmuka menambah pesan nama organisasi.....	93
Gambar 5.17 Implementasi antarmuka login	94
Gambar 5.18 Implementasi antarmuka melihat monitoring kerja.....	94
Gambar 5.19 Implementasi antarmuka melihat akun.....	95
Gambar 5.20 Implementasi antarmuka membuat akun	95
Gambar 6.1 <i>Flow graph</i> fungsi tambahOrder	97
Gambar 6.2 <i>Flow graph</i> fungsi work.....	99
Gambar 6.3 <i>Flow graph</i> fungsi work.....	103
Gambar 6.4 <i>Flow graph</i> fungsi tambahOrder	107

DAFTAR LAMPIRAN

LAMPIRAN A SURAT BUKTI ELISITASI KEBUTUHAN	150
LAMPIRAN B HASIL WAWANCARA	151
LAMPIRAN C KUESIONER <i>SYSTEM USABILITY SCALE</i>	152
LAMPIRAN D RESPONDEN <i>SYSTEM USABILITY SCALE</i>	155



BAB 1 PENDAHULUAN

1.1 Latar belakang

Berdasarkan pengertian pada Undang-Undang No. 30 Tahun 2004 tentang Jabatan Notaris dalam Bab I Ketentuan Umum dalam Pasal 1 disebutkan pengertian Notaris adalah pejabat umum yang berwenang untuk membuat akta otentik dan kewenangan lainnya sebagaimana dimaksud dalam Undang-Undang ini. Salah satu kantor notaris yang berkedudukan di Jakarta timur, yaitu Kantor Notaris Dewi Andriani S.H., M.H. yang telah buka praktek sejak tahun 2002 sejak diangkat berdasarkan SK Menteri Kehakiman Republik Indonesia, yaitu sejak tahun 2012.

Kantor notaris ini memiliki akta perbulannya kurang lebih tiga puluh sampai lima puluh buah akta masuk perbulannya. Akta tersebut akan disimpan sebagai protokol notaris. Penyimpanan protokol notaris dibutuhkan proses kehati-hatian karena protokol tersebut dapat tercecer, hilang, dan rusak. Selain itu, kelalaian dan keterbatasan karyawan yang diberikan tugas untuk dalam menyimpan protokol notaris dapat menjadi faktor yang harus diperhatikan. Apabila proses kehati-hatian ini tidak dijalankan dengan baik, notaris dapat dikenakan sanksi dari pasal 16 ayat (11) UUJN 2/2014, yaitu notaris dapat diberikan sanksi berupa peringatan tertulis, pemberhentian sementara, dan pemberhentian dengan hormat atau dengan tidak hormat. Oleh karenanya dibutuhkan penyimpanan protokol notaris secara digital dengan katalog penyimpanan yang disesuaikan dengan penyimpanan protokol fisik.

Selain itu dengan banyaknya penyimpanan protokol menimbulkan kesulitan bagi notaris apabila sewaktu-waktu klien notaris meminta perkembangan aktanya. Hal ini terjadi karena kantor ini masih menggunakan cara manual dan belum terkomputerisasi, sehingga seringkali Ibu Dewi Andriani, S.H., M.H. atau karyawan notaris terlambat untuk menjawab permintaan perkembangan akta klien notaris. Akta yang diminta klien harus dicari ke ruang penyimpanan terlebih dahulu yang kemudian mereka harus membuka lemari penyimpanan satu persatu dan memperkirakan dimana akta tersebut disimpan. Selain memperkirakan dimana akta tersebut disimpan, pemilik notaris dan pegawai notaris harus memperkirakan kapan akta tersebut dibuat. Karena itu untuk melihat perkembangan dari akta dari klien tersebut memakan waktu serta berkemungkinan kecil mencari dan mendapatkan akta tersebut dengan cepat. Pemilik kantor notaris, yaitu Ibu Dewi Andriani, S.H., M.H. menginginkan adanya sistem pengawasan pekerjaan kantor agar dapat mengetahui pekerjaan apa saja yang belum selesai, perkembangan dari pekerjaan yang sedang dikerjakan, dan siapa yang mengerjakan. Selain sistem pengawasan pekerjaan kantor, pemilik juga menginginkan adanya pengingat tenggat pekerjaan. Pengingat tenggat pekerjaan dibutuhkan karena pegawai kantor notaris terkadang lupa pekerjaan mana yang harus didahulukan, sehingga pekerjaan tidak selesai sesuai dengan jadwal yang telah dijanjikan.

Kemudian, protokol notaris yang telah disimpan tentu dapat mengalami kerusakan. Hal ini disebabkan oleh umur kertas yang hanya dapat bertahan beberapa belas tahun. Kertas yang sudah tua dapat menyebabkan mudah rusak karena menguning, lapuk, dan rapuh. Selain itu, kertas juga dapat termakan oleh rayap, atau bahkan hilang karena bencana alam yang menimpa di daerah tempat kedudukan kantor notaris yang bersangkutan. Seperti yang terjadi pada tahun 2004 lalu notaris di Aceh yang kehilangan dokumen-dokumen protokol tersebut dikarenakan terkena tsunami.

Berdasarkan permasalahan tersebut, dapat diketahui kantor notaris Dewi Andriani, S.H., M.H. sistem pembukuan dan sistem administrasi kerja masih bersifat konvensional sehingga belum dapat terorganisir secara baik. Agar dapat mengurangi permasalahan tersebut, maka dibuat sistem pembukuan dan administrasi protokol notaris pada Kantor Notaris Dewi Andriani, S.H., M.H. yang dapat menyimpan protokol notaris secara digital yang disesuaikan katalog terhadap penyimpanan fisik di kantor notaris, memberikan notifikasi umur kertas berdasarkan lama penyimpanan protokol notaris, pengingat tenggat pekerjaan dan pengawasan pekerjaan kantor, seperti pekerjaan apa saja yang belum selesai, perkembangan dari pekerjaan yang sedang dikerjakan, dan siapa yang mengerjakan.

1.2 Rumusan masalah

Berdasarkan latar belakang yang sudah dipeberkan di atas bahwa didapatkan rumusan masalah sebagai berikut:

1. Bagaimana hasil analisis kebutuhan Sistem Pembukuan dan Administrasi Protokol Notaris?
2. Bagaimana hasil rancangan Sistem Pembukuan dan Administrasi Protokol Notaris?
3. Bagaimana hasil implementasi Sistem Pembukuan dan Administrasi Protokol Notaris?
4. Bagaimana hasil pengujian dari Sistem Pembukuan dan Administrasi Protokol Notaris?

1.3 Tujuan

Penelitian berikut mempunyai tujuan sebagaimana, yakni:

1. Menganalisis kebutuhan dari Sistem Pembukuan dan Administrasi Protokol Notaris.
2. Merancang Sistem Pembukuan dan Administrasi Protokol Notaris sesuai analisis kebutuhan.
3. Membuat Sistem Pembukuan dan Administrasi Protokol Notaris.
4. Menguji Sistem Pembukuan dan Administrasi Protokol Notaris.

1.4 Manfaat

Dalam membangun Sistem Pembukuan dan Administrasi Protokol Notaris. Penulis memiliki harapan dalam penelitian ini memiliki manfaat yang dapat diambil bagi kantor notaris Dewi Andriani, S.H., M.H beserta. Manfaat utama dari sistem adalah:

Bagi kantor notaris Dewi Andriani, S.H., M.H.:

1. Dari aspek ekonomi, kantor notaris penyimpanan protokol notaris dapat lebih praktis dan efisien.
2. Dari aspek hukum, dapat membantu dan memudahkan dalam proses hukum mengenai hukum pembuktian, yaitu sebagai alat bukti cadangan secara elektronik.
3. Kantor Notaris dapat mengurangi resiko yang tidak terduga, yaitu kebakaran, digigit tikus atau serangga lain, dan force majeure, seperti bencana banjir.
4. Kantor Notaris dapat mengurangi resiko keterbatasan manusia yang mengakibatkan *human error*, seperti keterbatasan daya ingat dan kesalahan manajerial.

Bagi klien kantor notaris Dewi Andriani, S.H., M.H.:

1. Akta yang dimiliki oleh klien dapat lebih terjaga dari resiko penyimpanan protokol konvensional.
2. Mempercepat permintaan klien jika sewaktu-waktu meminta mencari aktanya.

1.5 Batasan masalah

Agar tidak terjadi deviasi terhadap hasil yang direncanakan, penelitian ini mempunyai batasan masalah, yakni:

1. Sistem dikonstruksikan pada lingkup Kantor Notaris Dewi Andriani, S.H., M.H.
2. Sistem akan dikonstruksikan menerapkan platform web.
3. Sistem yang akan dikonstruksikan digunakan kode pemrograman PHP, CSS, HTML5, dan *Javascript*.
4. Sistem yang akan dikonstruksikan digunakan framework Codeigniter.
5. Basis data yang akan dibangun dipakai basis data MySQL.
6. Sistem hanya dapat diakses oleh pimpinan notaris dan staff notaris.
7. Dibutuhkan konektivitas internet agar dapat melakukan penerimaan dan pengiriman data terhadap server yang tersedia.

1.6 Sistematika pembahasan

Bagian-bagian ringkas pada sistematika pembahasan deskripsi yang digunakan pada penelitian ini ialah:

BAB I PENDAHULUAN

Bab ini tercantum latar belakang masalah dari penelitian, rumusan masalah pada penelitian, tujuan dari penulisan, batasan masalah terhadap penelitian, manfaat penelitian sampai dengan sistematika penulisan.

BAB II LANDASAN KEPUSTAKAAN

Bab ini tercantum tentang paham dari penelitian yang telah dilakukan sebelumnya dan akan digunakan dalam melandasi penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini diutarakan metode penelitian yang menguraikan cara dan strategi operasi yang nantinya akan dipakai terhadap dibangunnya sistem.

BAB IV ANALISIS KEBUTUHAN

Bab ini menerangkan dan menguraikan analisis kebutuhan dan perancangan dari sistem yang perihal dibuat nantinya.

BAB V PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini diuraikan operasi perancangan sistem yang hendak dibuat dengan didasarkan analisis kebutuhan yang telah didapatkan atas suatu acuan, yaitu dari bab sebelumnya.

BAB VI PENGUJIAN

Bab ini diutarakan operasi pengujian dan analisis hasil pengujian kepada sistem dibangunnya sistem ini nantinya.

BAB VII PENUTUP

Bab ini menguraikan konklusi yang didapatkan dari produk pada perancangan sistem dan saran bagi pengembangan kian lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Dalam rujukan berisi penjelasan riset yang telah dikerjakan sebelumnya dan menggunakan lingkup yang serupa digunakan kajian pustaka sebagai rujukan. Diperoleh tiga penelitian yang dipakai untuk patokan dalam menjabarkan kebutuhan sistem. Penelitian pertama ialah studi yang telah dijalankan oleh Madina Shah. Judul penelitian ini memiliki judul “Impact of Management Information Systems (MIS) on School Administration: What Literature Says”. Penelitian ini mengulas berbagai penelitian yang menggunakan Management Information System (MIS) pada administrasi di sekolah. Dari penelitian ini dapat ditarik kesimpulan, yaitu penggunaan TIK di bidang manajemen pendidikan menunjukkan dampak yang sangat positif. Kemampuan kepala sekolah dan guru yang bekerja menggunakan TIK telah berkembang secara signifikan dari tahun ke tahun mereka menggunakan TIK untuk mendukung berbagai kegiatan administrasi di tingkat kelas dan sekolah (Shah, 2014).

Penelitian kedua berjudul “E-Commerce For Notaries In England & Wales”. Penelitian ini dikerjakan oleh Michael Lightowler. Pada penelitian ini dijelaskan pada “The 2006 Project” yang dikembangkan melalui pertemuan antara The Notaries Society dan salah satu mitra CDS (Certified Document Services) Adobe menggunakan Adobe Acrobat pdf untuk membuat Layanan Dokumen Bersertifikat untuk Notaris disebutkan bahwa notaris dapat memiliki kemampuan untuk mengesahkan segel elektronik dan tanda tangan mereka ke salinan elektronik (pdf) dari dokumen yang merupakan solusi untuk keberhasilan layanan ini. Kemudian, kemampuan untuk memindai dokumen yang ditandatangani yang dilakukan oleh para pelaku dan dapat menghasilkan salinan digital dalam format dokumen standar. Kumpulan dokumen dapat disimpan secara otomatis di *file* Notaris dan dapat dijadikan *backup* dengan cara biasa (Lightowler, 2007).

Penelitian ketiga adalah tesis yang dilakukan oleh Mohamat Riza Kuswanto dengan judul “Urgensi Penyimpanan Protokol Notaris dalam Bentuk Elektronik dan Kepastian Hukumnya di Indonesia”. Dalam penelitian ini menjelaskan bahwa dengan adanya ide atau wacana tentang pengalihan protokol notaris dari konvensional yang (paper based) menjadi elektronik (digital based) maka profesi notaris dalam menjalankan wewenang dan kewajibannya untuk menyimpan arsip-arsip dan dokumen-dokumen dalam protokol notaris tersebut akan menjadi lebih efektif dan efisien. Dikatakan efektif karena dokumen yang berbentuk elektronik tersebut mudah untuk diketemukan kembali apabila ada klien yang membutuhkan dokumen-dokumen tersebut dalam rentang waktu yang lama, selain itu pekerjaan notaris lebih efisien karena lebih ekonomis tidak membutuhkan banyak kertas dan lebih menghemat waktu bagi notaris dalam memasukkan dan menyimpan dokumen tersebut. Ide atau wacana pengalihan protokol notaris dalam bentuk elektronik terkendala dengan belum adanya undang-undang dan peraturan yang mengatur tentang keabsahan dari penyimpanan protokol notaris dalam bentuk

elektronik tersebut. Dengan demikian kekuatan pembuktian arsip-arsip dan dokumendokumen pada protokol notaris elektronik belum dapat disandingkan atau tidak dapat disetarakan dengan alat bukti autentik dengan kekuatan alat bukti yang berbentuk kertas tanpa ada alat bukti lainnya, contohnya adalah saksi ahli atau keterangan saksi (Mohamat Riza Kuswanto, 2017).

2.2 Rekayasa Perangkat Lunak

Menurut (Pressman, 2001) agar dapat mendapatkan perangkat lunak yang hemat, *trustable*, dan berjalan dengan tepat terhadap mesin dibutuhkan penyusunan dan penerapan prinsip-prinsip rekayasa perangkat lunak. Rekayasa perangkat lunak ialah ilmu yang memiliki keterkaitan terhadap sudut pabrikan perangkat lunak mulai bagian awal perincian sistem sampai dengan memelihara selepas sistem telah dipakai (Sommerville, 2011).

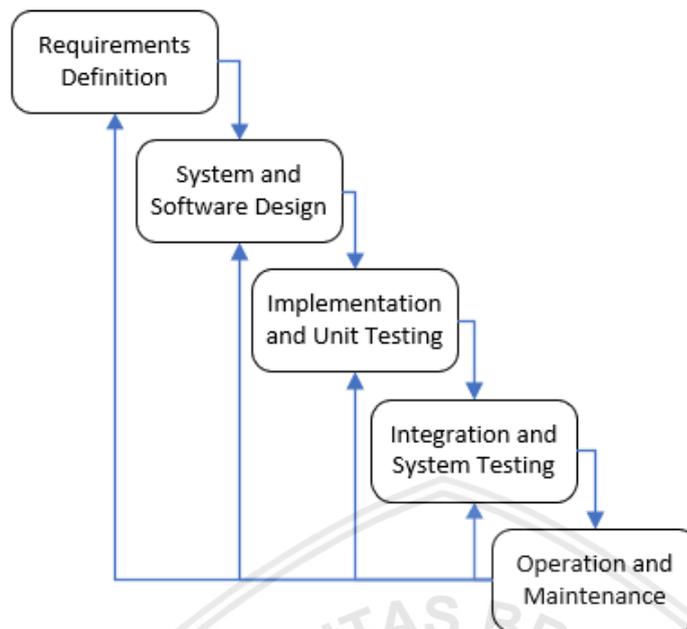
Proses perangkat lunak ialah pendekatan terstruktur yang dipakai terhadap rekayasa perangkat lunak (Sommerville, 2011). Proses perangkat lunak merupakan sekumpulan tahapan dalam menggarap produk perangkat lunak. Proses perangkat lunak memiliki empat tahap primer yang lazim. Pertama, spesifikasi perangkat lunak, yaitu tahap yang memastikan perangkat lunak yang akan dibangun bersama batas operasinya. Kedua, pengembangan perangkat lunak, yaitu tahap mendesain dan menciptakan perangkat lunak. Ketiga, validasi perangkat lunak, yakni tahap pengujian perangkat lunak untuk menentukan keselarasannya terhadap kebutuhan pengguna. Keempat, evolusi perangkat lunak, tahap dimana transisi dari perangkat lunak agar menyelaraskan dengan transisi kebutuhan user.

2.3 Pengembangan Perangkat Lunak

SDLC (*Software Development Life Cycle*) ialah prosedur yang dipakai di industri perangkat lunak agar merancang, mengelaborasi dan mengecek kapasitas perangkat lunak. SDLC mempunyai sejumlah tipe yang dapat dijalankan dalam pembangunan sistem, seperti: tipe *Waterfall*, tipe *Spiral*, tipe formal, dan tipe *extreme programming*. Pada pembangunan perangkat lunak ini, metode yang dipakai ialah metode *waterfall*. SDLC bertujuan agar dapat memberikan kapasitas yang tinggi dan sepadan, terlebih lagi dapat mengungguli ekspektasi pengguna. Selain itu, SDLC juga bertujuan agar perangkat lunak dapat dikerjakan waktu dan dana yang telah ditentukan (Sharma & Singh, 2016).

2.3.1 Metode *Waterfall*

Metode *waterfall* memberikan pendekatan siklus perangkat lunak secara teratur. Siklus dari metode ini ialah analisis, desain, implementasi, pengujian, dan fase pendukung (Rosa & Shalahuddin, 2011). Metode *waterfall* mempunyai sejumlah fase untuk dilakukan dilakukan (Sommerville, 2001). Metode *waterfall* merupakan cara yang dapat dibilang klasik yang berkarakter terstruktur, berurutan terhadap perangkat lunak. Tahapan metode *waterfall* ditunjukkan dalam Gambar 2.1:



Gambar 2.1 Model *waterfall*

Sumber: (Sommerville, 2011)

1. Analisis Kebutuhan

Pada tahap pertama dari model *waterfall*, yaitu *requirement*, pengembang perangkat lunak mencari kebutuhan dari keseluruhan sistem yang akan dibuat. Pengembang akan melakukan komunikasi dengan pengguna agar dapat memahami apa yang telah menjadi harapan pengguna, memahami semua kebutuhan yang diperlukan dari pengguna, dan batas-batas apa saja ketika menciptakan perangkat lunak tersebut. *Requirement* dapat dimanfaatkan dengan wawancara, diskusi, dan survei. Selepas mendapatkan informasi, pengembang perangkat lunak bisa membedah kebutuhan yang tersedia untuk perangkat lunak.

2. Perancangan Sistem dan perangkat lunak

Pada tahap desain sistem, hasil dari *requirement* yang sudah didapatkan terhadap tahap sebelumnya diubah menjadi representasi ke dalam desain atau cetak biru dari perangkat lunak. Desain yang telah dibuat akan diimplementasikan pada tahap berikutnya.

3. Implementasi dan pengujian unit

Pada tahap implementasi sistem, akan dibuat berdasarkan desain sistem yang telah dibuat pada tahap sebelumnya. Dan pengujian unit akan memastikan setiap program telah sesuai dengan rincian detailnya.

4. Pengujian integrasi dan sistem

Tahap ini program digabungkan menjadi menjadi sebuah sistem yang utuh dan lengkap. Kesesuaian suatu kode program dan kebutuhan merupakan tujuan dari pengujian ini. Perangkat lunak akan diserahkan kepada pengguna setelah diuji.

5. Operasi dan perawatan (belum pasti dikerjakan)

Pada tahap paling akhir dari metode *waterfall*, perangkat lunak telah dibuat dan telah diuji akan masuk kedalam tahap *maintenance* atau pemeliharaan. Pemeliharaan dari sistem dijalankan agar menaikkan kapasitas dan keandalan dari sistem yang telah dibangun. Pemeliharaan terlibat mengoreksi kesalahan dan kegagalan yang tidak didapati terhadap tahap *integration & testing*.

Keunggulan pada metode ini, yakni:

1. Pelaksanaannya sederhana.
2. Kualitas dari perangkat lunak yang dihasilkan akan lebih baik dikarenakan dalam pelaksanaan metode ini dilakukan secara bertahap, sehingga tidak terfokus terhadap satu tahap tertentu.
3. Dokumentasi dari pengembangan perangkat lunak akan terorganisir baik karena metode ini diharuskan setiap tahap harus terselesaikan dengan lengkap melangkah ke tahap selanjutnya.
4. Bisa meminimalisir kekeliruan yang tampaknya timbul karena proses pengembangan metode ini merupakan fase satu per satu.

Kekurangan pada metode waterfall yakni:

5. Metode yang bersifat kaku menyebabkan sulit untuk melangsungkan peralihan apabila terjadi kekeliruan di dalam tahapan metode ini.
6. Sulit untuk kembali ke tahap sebelumnya dan mengubah segala sesuatu yang tidak terdokumentasi dengan baik.

2.3.2 Pendekatan Berorientasi Objek

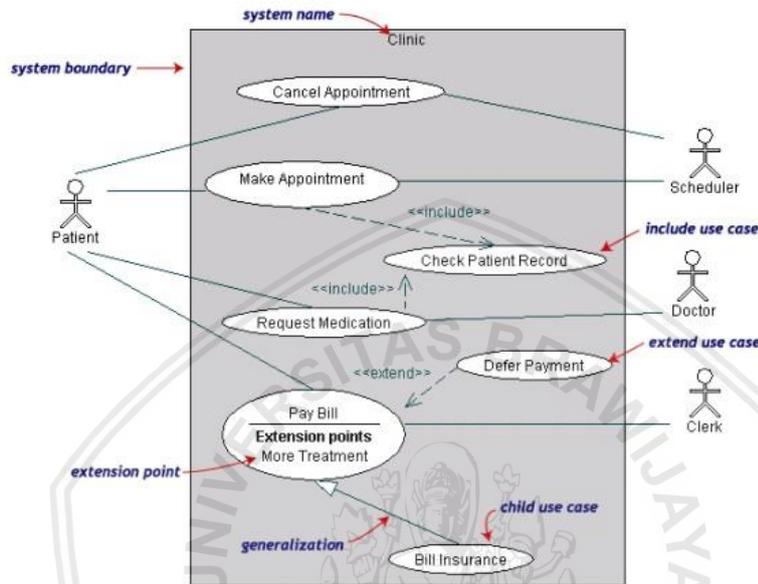
Pendekatan berorientasi objek ialah sebuah paradigma pemrograman berdasarkan objek. Dalam paradigma pemrograman ini, semua data dan fungsi akan dibungkus ke dalam kelas atau objek. Setiap objek akan dapat menerima pesan, memproses data, mengirim, menyimpan, dan memanipulasi data. Objek dapat berinteraksi satu sama lain dengan cara saling memberikan informasi-informasi satu sama lain.

2.3.3 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) ialah ragam umum untuk mendesain, memvisualisasi, dan penyimpanan sistem dari perangkat lunak (Dharwiyanti & Wahono, 2003). Agar dalam operasi peninjauan dan penerimaan kode program terhadap sistem dijalankan lebih sederhana, UML digunakan dengan cara memvisualisasikan dokumentasi UML. Kemudian, UML dipakai untuk membangun model dan memudahkan operasi pengembangan perangkat lunak. Selain itu, UML dapat dipakai selaku *tools* dalam memindahkan pengetahuan antara pengembang dan sistem.

2.3.3.1 Use Case Diagram

Fungsionalitas dari suatu sistem digambarkan oleh *use case diagram*. Selain itu, dipakai untuk menggambarkan hubungan yang terjalin antara aktor dan sistem. Dengan *use case diagram*, dalam merumuskan kebutuhan dari suatu sistem dapat jalankan lebih sederhana. (Dharwiyanti & Wahono, 2003). Berikut ialah representatif *use case diagram* dalam Gambar 2.2.



Gambar 2.2 Use case diagram

Sumber: (Dharwiyanti & Wahono, 2003)

2.3.3.2 Sequence Diagram

Sequence diagram dimanfaatkan untuk visualisasi dari sinopsis yang telah dijalankan oleh objek lewat *lifetime* objek, *message* yang dikirim, dan diterima untuk objek. Selain itu, *Sequence diagram* ialah visualisasi hubungan antara entitas yang didalam atau seputar sistem, sebagaimana tampilan, *user*, dan lainnya (Dharwiyanti & Wahono, 2003). Elemen pada *sequence diagram* adalah:

Tabel 2.1 Elemen *sequence diagram*

Gambar	Keterangan
	<i>Entity class</i> ialah unit kelas yang selaku suatu dasar untuk menyusun basis data.
	<i>Boundary class</i> ialah unit kelas yang selaku bentuk antara sistem dan aktor.
	<i>Control class</i> ialah unit yang berisikan logika selaku jembatan antara <i>entity</i> dan <i>boundary class</i> .

Tabel 2.1 Elemen sequence diagram (lanjutan)

Gambar	Keterangan
	<i>Message</i> ialah tanda dalam melayangkan pesan antar objek.
	<i>Recursive</i> ialah tanda dalam menjalankan melayangkan pesan terhadap pesan itu secara mandiri.
	<i>Activation</i> ialah visualisasi dalam eksekusi suatu aktivitas dari objek.

2.3.3.3 Class Diagram

Untuk memvisualisasikan antara hubungan antara kelas terhadap kelas lainnya yang lalu bagian dari kelas tersebut dimanfaatkan *class diagram* (Hendini, 2016). Selain itu, bagian kelas seperti atribut dan operasi yang terdapat didalamnya juga divisualisasikan pada *class diagram*. *Class diagram* terbagi atas nama kelas, atribut, operasi, dan relasi. Relasi antar-kelas terbagi atas asosiasi, asosiasi berarah, generalisasi, kebergantungan, dan agregasi.

2.4 Sistem Pembukuan Protokol Notaris

Sistem pembukuan merupakan pencatatan transaksi keuangan. Transaksi yang dimaksud meliputi penjualan, pembelian, pendapatan, dan pengeluaran oleh suatu perseorangan atau organisasi. Dengan pembukuan, perusahaan dapat melacak semua informasi pada buku-buku perusahaan agar dapat membuat keputusan operasional, investasi, dan keuangan yang penting. Dalam pembukuan protokol notaris, arsip-arsip dalam bentuk akta minuta dijilid dalam bentuk bundel kemudian akan disimpan ke dalam bentuk protokol notaris.

2.5 Sistem Administrasi Protokol Notaris

Sistem administrasi merupakan segala sesuatu bentuk kegiatan tata usaha. Ketatausahaan adalah kegiatan yang penting dalam suatu pekerjaan yang sedang dilakukan dan tujuan dari suatu organisasi. Khususnya hal yang menyangkut dengan peranan informasi sebagai darah dari suatu organisasi. Dalam pengertian yang demikian administrasi biasanya hanya dikaitkan dengan kegiatan-kegiatan ketatausahaan yang mencakup korespondensi, kesekretariatan, penyusunan laporan, dan kearsipan (Siagian, 2001: 267).

2.6 Profesi Notaris

Profesi notaris adalah sebuah sebutan profesi untuk seseorang yang telah mendapatkan pendidikan hukum yang dilisensi oleh pemerintah untuk melakukan hal-hal hukum, khususnya sebagai saksi penandatanganan pada dokumen. Bentuk profesi notaris berbeda-beda tergantung pada sistem hukum. Notaris merupakan salah satu cabang dari profesi hukum yang tertua di dunia.

Jabatan notaris tidak ditempatkan di lembaga eksekutif, legislatif, dan yudikatif. Notaris diamanahkan untuk memiliki posisi netral, sehingga ketika ditempatkan di salah satu dari ketiga badan negara tersebut maka notaris tidak lagi dapat dianggap netral. Dengan posisi netral tersebut, notaris diharapkan dapat memberikan penyuluhan hukum untuk dan atas tindakan hukum yang dilakukan notaris berdasarkan permintaan kliennya. Dalam hal melakukan tindakan hukum untuk kliennya, notaris juga tidak boleh memihak kliennya, karena tugas notaris adalah untuk mencegah terjadinya masalah.

2.7 Protokol Notaris

Protokol notaris ialah suatu kumpulan dokumen dimiliki oleh arsip negara yang kemudian wajib disimpan dan dipelihara oleh notaris. Protokol Notaris terdiri:

1. Akta orisinal/minuta akta.
2. Buku daftar akta ataupun repertorium.
3. Buku daftar nama-nama penghadap atau *klapper*.
4. Buku daftar akta dibawah tangan yang penandatngannya dilakukan terhadap notaris/akta dibawah tangan yang telah didaftarkan.
5. Buku daftar wasiat.
6. Buku daftar protes.
7. Buku daftar lainnya yang wajib disimpan oleh notaris yang didasarkan peraturan perundang-undangan.

Jika Notaris telah meninggal dunia, maka protokol notaris akan diserahkan terhadap notaris yang telah ditunjuk oleh Majelis Pengawas Daerah (MPD) yang dilakukan oleh ahli waris notaris yang meninggal tersebut. Kemudian apabila notaris telah wafat dalam menjalankan cuti, maka notaris pengganti wajib untuk serahkan protokol notaris kepada Majelis Pengawas Daerah (MPD).

2.8 Model-view-controller

Model view controller berfungsi untuk menjarakkan pembangunan aplikasi yang didasarkan elemen utama yang menciptakan suatu penerapan semacam manipulasi data, antarmuka user, dan elemen yang menjadikan *control* terhadap aplikasi. Tersimpul tiga kelompok komponen, yakni model, view dan controller. Model berkaitan kontan dengan basis data untuk mengatur data (*insert, update, delete dan search*), memproses validasi dari elemen controller. Tetapi tidak mampu berkaitan langsung atas elemen view. View ialah Elemen yang memproses logika presentasi. Elemen ini umumnya disusun oleh controller sedangkan view beroperasi dalam mendapatkan dan menggambarkan data terhadap user. Elemen ini tidak mempunyai jalan masuk secara kontan untuk model. Sedangkan controller dimana yang memproses interaksi antara model dan view. Kemudian controller beroperasi dalam mendapatkan *request* dan data dari user yang selanjutnya memutuskan apa yang akan nantinya diproses terhadap aplikasi (Riyandwyana & Mukhlason, 2012).

2.9 Codeigniter

Codeigniter ialah suatu framework php yang berkarakter terbuka dan mampu menyokong pengembang saat membuat suatu aplikasi web berbasis PHP (Hakim, 2010). Formula MVC, yaitu model, view, dan controller digunakan oleh *framework* ini. Framework codeigniter mempunyai maksud agar menyokong pengembang untuk membuat sebuah sistem berbasis web. Kelebihan dari *framework* ini, yakni penggunaannya gratis, sintaks terstruktur, dapat dijalankan dengan mudah, konfigurasinya tidak banyak, dan telah mempunyai *helper* dan *library*. Sedangkan untuk kekurangannya, yakni untuk web yang mempunyai skala besar kurang dapat digunakan oleh *framework* ini dan *library*nya memiliki batas tertentu.

2.10 Pengujian Perangkat Lunak

Pengujian perangkat lunak memiliki tujuan merancang beberapa kasus uji yang mempunyai kemungkinan tinggi untuk mendapatkan *error* yang ada pada sistem yang dikonstruksi (Pressman, 2010). Pengujian terhadap perangkat lunak memiliki tujuan dalam mencari dan menemukan *error*. Hal tersebut mampu ditemukan, seperti di konten, fungsi, usabilitas, navigasi, performa, kapasistas, dan keamanan. Metode yang dijalankan pada penelitian ini ialah *whitebox testing*, *blackbox testing*, dan *integration testing*. Untuk memastikan kemudahan pengguna dalam penggunaannya akan dilakukan pengujian *usability*.

2.10.1 Pengujian Unit

Pengujian unit ialah pengujian yang utama kali akan dikerjakan. Hal yang dicoba pada pengujian unit terhadap pemrograman berorientasi objek, yakni pengujian pada tingkatan kelas. Ciri dari pengujian unit ialah pengujian dijalankan di tingkatan unit secara tersekat dari komponen-komponen lain yang berada di sistem. Hal ini agar melindungi dampak dari komponen-komponen lain. (Spillner, Linz dan Schaefer, 2014). *Whitebox testing* ialah suatu cara dalam mengukur suatu perangkat lunak melalui cara memandang modul mampu diteliti dan menganalisis *coding* pada program yang sebelumnya telah dibuat terdapat kesalahan atau tidak.

Pressman (2010) menjelaskan apabila *whitebox testing* memanfaatkan struktur kontrol yang ialah bagian daripada tingkatan desain komponen untuk menghasilkan sebuah kasus uji. Dapat dikatakan, pengujian *white box* didesain apabila *source code* telah dikonstruksi. Kasus uji yang diproduksi, akan memanfaatkan metode *whitebox testing* mengamankan segenap jalur independen yang tersedia suatu modul. Hal ini mampu digunakan minimal satu kali, mengecek seluruh keputusan logis dan akan menyampaikan hasilnya dengan kesesuaian bernilai benar atau salah berlandaskan konteks yang telah didapat, memanfaatkan pemeriksaan pada seluruh *loop* melalui batas tertentu, dan melakukan percobaan pada struktur data internal agar memastikan apabila struktur itu memiliki validitas.

Whitebox testing memiliki dasar terhadap pengujian prosedur yang rinci. Jalur logis yang dijajal pada perangkat lunak dan kerja sama antara bagian yang dicobakan berlandaskan sejumlah kondisi tidak/dan perulangan tertentu. Dalam pengujian sistem akan dipakai *basis path testing*. *Basis path testing* membuahakan penghitungan kompleksitas logis dari desain prosedural. Penghitungan dapat dipakai dalam memastikan fondasi untuk kelompok jalur yang akan dieksekusi.

2.10.2 Pengujian Integrasi

Pengujian integrasi bermaksud agar menjamin bahwa seluruh komponen mampu berinteraksi secara baik dan benar sinkron dengan kebutuhan (Spillner, Linz, dan Schaefer, 2014). Jika ditemukan suatu kekurangan, pengujian integrasi dapat membuka kekurangan. Dari sisi cara berinteraksi antara bagian yang saling bersinggungan. Pada pengujian integrasi tersimpul dua strategi yang dapat dimanfaatkan, yakni integrase atas ke bawah dan integrase bawah ke atas. Strategi atas ke bawah pada pengujian akan dimulai dari komponen tingkatan atas (komponen utama), kemudian bagian tingkatan bawah akan memanggil bagian tingkatan bawah. Apabila bagian tingkatan bawah belum rampung dikonstruksi, kemudian pengujian dapat memanfaatkan sebuah *test stub*, yakni sebuah bagi yang dapat simulasikan bagian tingkatan bawah yang belum rampung dikonstruksi. Untuk strategi integrasi bawah ke atas pengujian, akan mulai dari bagian tingkatan bawah menuju bagian tingkatan atas (komponen utama). Jika bagian tingkatan atas belum rampung dikonstruksi maka pengujian dapat memanfaatkan *test driver*, yakni dapat dimanfaatkan untuk simulasikan bagian tingkatan atas yang belum dikonstruksi.

2.10.3 Pengujian Validasi

Salah satu pengujian yang wajib dilakukan sebelum sistem dimanfaatkan terhadap lingkungannya ialah pengujian validasi. Teknik *blackbox testing* ialah strategi pengujian yang dimanfaatkan agar dapat mengetahui keselarasan suatu fungsi tertentu dari suatu sistem. *Blackbox testing* ialah penghampiran yang ditujukan mencoba sistem berlandaskan spesifikasi tanpa harus menjalankan uji kepada kode program tersebut (Sommerville, 2011). *Blackbox testing* yang dikhususkan terhadap perangkat lunak. Pengujian *blackbox* dimanfaatkan agar mencari *error* saat kode dijalankan.

Blackbox testing akan mencari *error* pada beberapa kelompok (Pressman, 2010), yakni fungsi yang memiliki kesalahan atau fungsi yang tidak sempurna, antarmuka yang *error*, *error* ketika menjalankan akses basis data eksternal, *error* dalam *behaviour*, dan *error* ketika inisialisasi dan terminasi. *Blackbox testing* condong terhadap dilakukan atas tahap yang terakhir dikarenakan tidak melihat struktur control. *Black box testing* dilakukan dengan mempersiapkan sejumlah kasus uji yang telah didesain agar mendapatkan *error* di setiap tingkatann validasi perangkat lunak. Kasus uji yang tersedia, diharapkan mendapatkan *error* dengan potensial dalam waktu dan upaya seminimal mungkin. Kasus uji yang telah dibuat agar dapat dijalankan terhadap suatu fungsi yang ada. Kemudian kasus uji akan

divalidasi untuk mendapatkan apakah sistem sudah berfungsi penuh dan disaat waktu yang sama memulai pelacakan *error* yang dapat timbul.

2.10.4 Pengujian *Usability*

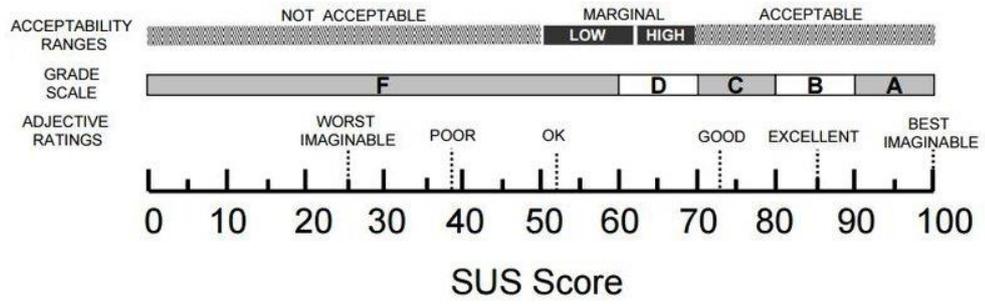
Pengujian *usability* digunakan dalam melakukan pengujian terhadap sistem menguji sistem secara langsung terhadap responden. Hal ini bertujuan agar mengetahui tingkatan kemudahan penggunaan terhadap sistem oleh responden. Setelah sistem dapat digunakan akan dilakukan pengujian ini, sehingga dijumlahkan penilaian dari hasil dan analisis yang dijumlah berlandaskan metode yang telah dimanfaatkan. Pengujian *usability* ini dimanfaatkan metode *System Usability Scale (SUS)*. Metode ini adalah pengujian dengan arah *user-oriented* yang menggunakan alat pengukuran yang cepat, kotor, dan dapat diandalkan (Brooke, 1986). Selain itu, responden pada metode ini akan diberikan peluang dalam menilai sistem tanpa diberikan pengarahan terlebih dahulu.

Metode SUS terdiri atas sepuluh pertanyaan menggunakan skala *likert* dengan poin nilai satu sampai lima. Keterangan pada setiap skala *likert* dapat dilihat pada Tabel 2.2. Sebelum mencari nilai akhir dengan mengalikan total poin terhadap 2.5 poin, pertanyaan dengan nomor 1, 3, 5, 7, dan 9 adalah pertanyaan yang memiliki arah ke hal positif sedangkan untuk pertanyaan dengan nomor 2, 4, 6, 8, dan 10 adalah pertanyaan yang memiliki arah ke hal negatif. Penilaian dilakukan dengan cara memberikan pengurangan satu poin nilai dari respon yang diberikan oleh responden terhadap pertanyaan dengan nomor 1, 3, 5, 7, dan 9 dan pengurangan lima nilai terhadap pertanyaan dengan nomor 2, 4, 6, 8, dan 10.

Tabel 2.2 Keterangan poin skala likert

No	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Pada poin akhir berada diantara *range* 0 dan 100. Kriteria skor poin dapat ditunjukkan dalam Gambar 2.3. Dimana poin simpulan tersebut dapat diketahui seberapa jumlah tingkatan usabilitas dan *acceptable* dari sistem yang telah dibuat. Penilaian ini memiliki tiga kelompok, yaitu *not acceptable*, *marginal*, dan *acceptable* dengan rentang poin masing-masing poin SUS 0 - 50.9, 51 – 70.9, dan 71 – 100.



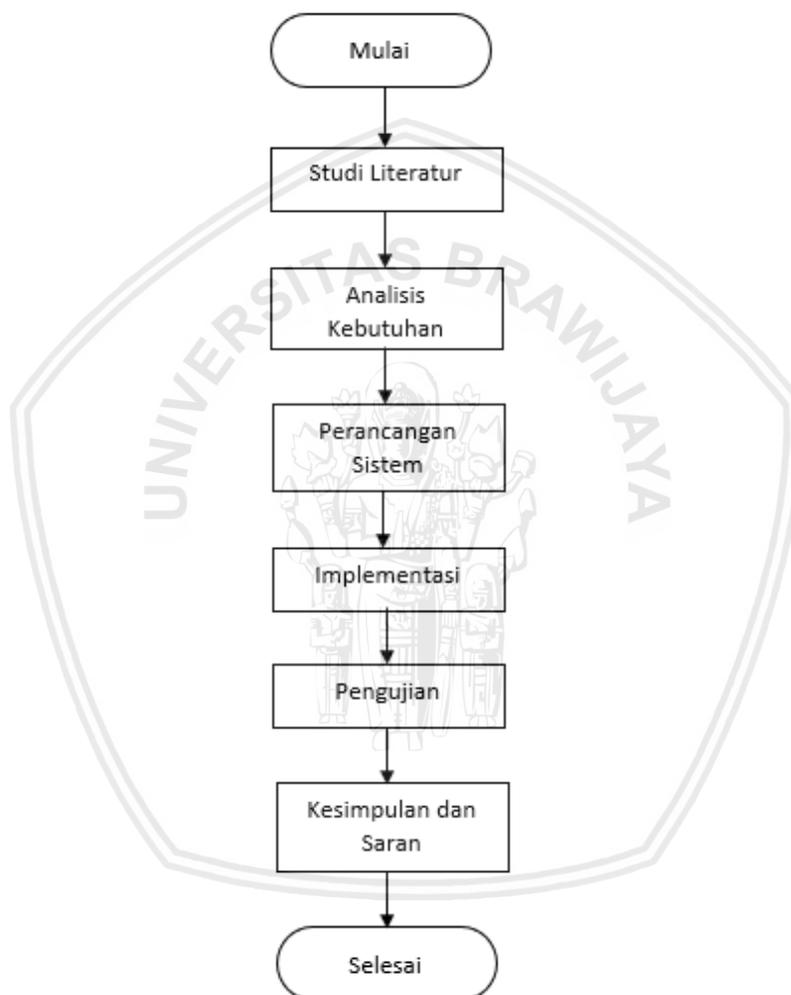
Gambar 2.3 Rating dan skala konversi poin rata-rata SUS

Sumber: (Bangor, Kortum, & Miller, 2009)



BAB 3 METODOLOGI PENELITIAN

Dalam bab ini akan dijelaskan langkah dari rencana penelitian yang digunakan dalam mengerjakan penelitian ini. Langkah dari rencana penelitian ini, yakni studi literatur, suatu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan beserta saran. Metode penelitian akan dimanfaatkan sebagai arahan dari pelaksanaan penelitian ini. Diagram alir metodologi penelitian yang digunakan di penelitian ini ditunjukkan dalam Gambar 3.1.



Gambar 3.1 Diagram alir penelitian

3.1 Studi Literatur

Penulisan pada penelitian ini digunakan studi literatur yang berisi pustaka acuan. Basis yang dimanfaatkan bersumber dari jurnal-jurnal ilmiah, buku, dan situs yang semuanya berkaitan terhadap proses penelitian ini. Teori dan pustaka dari penelitian ini, yaitu rekayasa perangkat lunak, pengembangan perangkat lunak, sistem pembukuan protokol notaris, sistem administrasi protokol notaris, profesi notaris, protokol notaris, *model view controller*, *codeigniter*, dan pengujian perangkat lunak.

3.2 Analisis Kebutuhan

Dalam menggali permasalahan dan mengidentifikasi kebutuhan apa yang harus ada pada sistem yang dibangun digunakan analisis kebutuhan. Tahap analisis kebutuhan sangat vital, mengingat identifikasi kebutuhan yang telah dilakukan akan mempengaruhi tahapan-tahapan yang ada.

Studi literatur, studi lapangan yang berbentuk observasi langsung, dan wawancara dibutuhkan dalam membangun sistem. Hal ini memiliki tujuantujuan, yakni mengetahui kebutuhan sistem yang dibutuhkan oleh pengguna. Hal ini dilakukan untuk meminimalisir baik kesalahan kecil, maupun kesalahan besar saat membangun sistem. Studi literatur dimanfaatkan selaku dasar acuan dengan menemukan referensi, seperti buku, jurnal, dan internet. Referensi-referensi ini dilakukan untuk pendukung pada sistem yang akan dibuat nantinya.

Untuk menspesifikasikan kebutuhan Kantor Notaris Dewi Andriani, S.H., M.H. yang dibutuhkan nantinya menggunakan produk dari studi literatur dan studi lapangan Kemudian akan diverifikasi dan divalidasi sesuai dengan kebutuhan oleh pihak kantor notaris. pengenalan aktor-aktor yang akan berpartisipasi pada sistem, rekognisi kebutuhan fungsional, kebutuhan non-fungsional, *use case diagram*, dan *use case scenario* ialah hasil dari tahapan ini.

3.3 Perancangan Sistem

Perancangan sistem ialah tahapan rancangan analisis kebutuhan yang telah dibuat terhadap sistem. Perancangan sistem akan menjadi dasar untuk membangun dan melakukan pengujian terhadap sistem. Perancangan *sequence diagram*, *class diagram*, algoritme, basis data, dan antar muka hasil merupakan hasil dari tahapan analisis kebutuhan.

3.4 Implementasi

Fase implementasi merupakan tahap yang digunakan setelah tahapan perancangan telah rampung. Setiap kebutuhan yang sudah dituliskan juga harus disepakati oleh pengguna. Fase implementasi adalah fase pembangunan sistem sesuai dengan perancangan yang telah dibuat. Proses pada fase implementasi, yakni implementasi dilakukan didasarkan oleh spesifikasi sistem dalam membangun situs ini. Lalu elemen, seperti bahasa pemrograman PHP dan *framework codeigniter* akan membentuk sistem berbasis web ini sebagai implementasi perangkat lunak. Kemudian implementasi basis data diimplementasikan dengan memanfaatkan *MySQL*, dan implementasi antarmuka didasarkan oleh perancangan antarmuka sistem.

3.5 Pengujian

Sistem yang telah dibangun dan spesifikasi kebutuhan yang sudah dianalisis terhadap sistem akan diperiksa dan akan ditemukan dengan pengujian. Pengujian unit dan pengujian validasi digunakan pada sistem ini. Pengujian unit adalah pengujian yang digunakan melalui mengecek elemen terkecil dari sistem. Selain

itu, akan dijalankan dengan metode *whitebox testing* yang dicek melalui memanfaatkan tipe *basis path testing*. Sedangkan untuk mengecek kesamaan antar kebutuhan yang sudah didefinisikan terhadap sistem yang dibuat akan menggunakan pengujian validasi. Tujuannya ialah mengetahui apakah sistem yang dibuat sudah tepat atau belum. Untuk menguji kemudahan saat menggunakan sistem yang akan digunakan setiap pengguna akan digunakan pengujian usabilitas.

3.6 Kesimpulan dan Saran

Ketika semua tahapan penggarapan penelitian telah rampung, kesimpulan dan saran dapat didapatkan. Kesimpulan diambil berdasarkan dari dari produk perancangan dan pengujian. Kesimpulan dibuat agar memenuhi rumusan masalah yang ada, sedangkan saran digunakan seperti masukan untuk membantu menyempurnakan pembangunan sistem kelak.



BAB 4 ANALISIS KEBUTUHAN

Tahap awal dalam melakukan pembangunan sistem akan digunakan analisis kebutuhan. Tujuan dari analisis kebutuhan ialah agar dapat menentukan kebutuhan yang harus ada pada Sistem Pembukuan dan Administrasi Protokol Notaris. Melalui hasil wawancara dan observasi didapatkan produk studi lapangan untuk kebutuhan Kantor Notaris Dewi Andriani, S.H., M.H. Selain itu, analisis kebutuhan akan menjabarkan aktor yang akan ikut serta terhadap sistem. Lalu kedudukan untuk aktor tersebut terhadap sistem dan hubungannya dengan perangkat lunak.

4.1 Deskripsi Sistem

Sistem Pembukuan dan Administrasi Protokol ialah sistem yang dibentuk atas sasaran utama agar menyokong memberikan kemudahan internal kantor notaris dalam hal pembukuan berkas notaris dan administrasi protokol notaris. Sistem ini dibangun berbasis. Secara umum, sistem ini dibuat kedalam lima bagian utama, yaitu:

1. Sistem administrasi pekerjaan dengan sistem disposisi
2. Pengawasan Kerja
3. Penyimpanan Protokol Notaris secara Digital
4. Notifikasi Tenggat Pekerjaan
5. Notifikasi Umur Kertas

4.2 Identifikasi Aktor

Kedudukan aktor yang berpartisipasi terhadap sistem akan diutarakan pada identifikasi aktor. Hasil observasi ketika studi lapangan akan menentukan aktor yang ada. Pada hasil observasi pada sistem ini, tersimpul dua kedudukan aktor yakni pemilik dan pegawai. Kolom nomor, aktor, dan deskripsi yang mendeskripsikan kegiatan masing-masing kegiatan dari aktor akan mengisi tabel identifikasi aktor. Identifikasi aktor ditunjukkan pada Tabel 4.1.

Tabel 4.1 Identifikasi aktor sistem

No.	Aktor	Deskripsi
1	Pengguna	Pengguna ialah individu yang belum teridentifikasi oleh sistem.
2	Pemilik	Pemilik ialah pemilik kantor Notaris Dewi Andriani yang mempunyai hak akses pada

Tabel 4.1 Identifikasi aktor sistem (lanjutan)

No.	Aktor	Deskripsi
2	Pemilik	seluruh fungsi yang terdapat di dalam sistem.
3	Pegawai	Pegawai adalah individu yang berbuat kepada pemilik kantor notaris yang mempunyai akses terbatas tertentu di dalam sistem.

4.3 Kebutuhan Fungsional

Kebutuhan yang harus disediakan sistem dan mampu digunakan oleh pengguna ialah kebutuhan fungsional. Kebutuhan fungsional yang dibutuhkan oleh pengguna dihasilkan oleh elisitasi kebutuhan yang telah dikerjakan. Kebutuhan fungsional sistem ini mempunyai total 62 kebutuhan. Nomor, *use case*, aktor, spesifikasi, dan kode akan mengisi tabel kebutuhan fungsional. Pada Tabel 4.2. ditunjukkan tabel kebutuhan fungsional.

Tabel 4.2 Kebutuhan fungsional

No.	<i>Use Case</i>	Aktor	Spesifikasi	Kode
1	<i>Login</i>	Pengguna	Sistem harus dapat mengidentifikasi pengguna sebagai pegawai atau pemilik.	EPN_F_1
2	<i>Logout</i>	Pemilik, Pegawai	Sistem harus dapat mengeluarkan aktor pegawai atau pemilik dari sistem sebagai pengguna.	EPN_F_2
3	Melihat Akun	Pemilik	Sistem harus dapat menampilkan data akun yang telah terdaftar di database.	EPN_F_3
4	Menambah Akun	Pemilik	Sistem harus dapat menambahkan akun.	EPN_F_4
5	Mengubah Akun	Pemilik	Sistem harus dapat mengubah akun.	EPN_F_5
6	Menghapus akun	Pemilik	Sistem harus dapat menghapus akun.	EPN_F_6
7	Menambah Klien	Pegawai	Sistem harus dapat menambahkan klien.	EPN_F_7
8	Melihat Klien	Pemilik, Pegawai	Sistem harus dapat menampilkan data klien.	EPN_F_8
9	Mengubah Klien	Pemilik, Pegawai	Sistem harus dapat mengubah klien.	EPN_F_9

Tabel 4.2 Kebutuhan fungsional (lanjutan)

10	Melihat Riwayat Klien	Pemilik, Pegawai	Sistem harus dapat melihat riwayat klien	EPN_F_10
11	Melihat Pesan Nama Organisasi	Pemilik, Pegawai	Sistem harus dapat melihat pesan nama organisasi	EPN_F_11
12	Menambah Pesan Nama organisasi	Pegawai	Sistem harus dapat menambahkan pesan nama organisasi.	EPN_F_12
13	Melihat Rincian Berkas Klien pada Pesan Nama Organisasi	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada pesan nama organisasi.	EPN_F_13
14	Mengubah Pesan Nama organisasi	Pegawai	Sistem harus dapat mengubah pesan nama organisasi.	EPN_F_14
15	Menolak Pesan Nama organisasi	Pemilik	Sistem harus dapat menolak pesan nama organisasi.	EPN_F_15
16	Menghapus Pesan Nama Organisasi	Pemilik	Sistem harus dapat menghapus pesan nama organisasi.	EPN_F_16
17	Memverifikasi Pesan Nama Organisasi	Pemilik	Sistem harus dapat memverifikasi pesan nama organisasi.	EPN_F_17
18	Melihat Buat Akta Organisasi	Pemilik, Pegawai	Sistem harus dapat melihat buat akta organisasi.	EPN_F_18
19	Melihat Rincian Berkas Klien pada Buat Akta Organisasi	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada buat akta organisasi.	EPN_F_19
20	Membarui Status Buat akta Organisasi	Pegawai	Sistem harus dapat membarui status akta organisasi.	EPN_F_20
21	Mengonfirmasi Status Buat Akta Organisasi Selesai	Pegawai	Sistem harus dapat mengonfirmasi akta organisasi selesai.	EPN_F_21

Tabel 4.2 Kebutuhan fungsional (lanjutan)

No.	Use Case	Aktor	Spesifikasi	Kode
22	Melihat Pendaftaran SK Organisasi	Pemilik, Pegawai	Sistem harus dapat menampilkan data pendaftaran SK organisasi.	EPN_F_22
23	Mengunggah Berkas SK Organisasi	Pemilik	Sistem harus dapat mengunggah berkas SK organisasi.	EPN_F_23
24	Melihat Cek Nama Wasiat	Pemilik, Pegawai	Sistem harus dapat melihat cek nama wasiat.	EPN_F_24
25	Melihat Rincian Berkas Klien pada Pendaftaran SK Organisasi	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada pendaftaran SK organisasi.	EPN_F_25
26	Menambah Cek Nama Wasiat	Pegawai	Sistem harus dapat menambahkan cek nama wasiat.	EPN_F_26
27	Menambah Cek Nama Wasiat	Pegawai	Sistem harus dapat menambahkan cek nama wasiat.	EPN_F_27
28	Melihat Rincian Berkas Klien Pada Cek Nama Wasiat	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada cek nama wasiat.	EPN_F_28
29	Memverifikasi Cek Nama Wasiat	Pemilik	Sistem harus dapat memverifikasi cek nama wasiat.	EPN_F_29
30	Menolak Cek Akta Wasiat	Pemilik	Sistem harus dapat menolak cek akta wasiat.	EPN_F_30
31	Mengubah Buat Akta Wasiat	Pegawai	Sistem harus dapat mengubah buat akta wasiat.	EPN_F_31
32	Memverifikasi Status Pewasiat	Pemilik	Sistem harus dapat memverifikasi status pewasiat.	EPN_F_32
33	Melihat Buat Akta Wasiat	Pemilik, Pegawai	Sistem harus dapat melihat buat akta wasiat.	EPN_F_33
34	Melihat Rincian Berkas Klien pada Buat Akta Wasiat	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada buat akta wasiat.	EPN_F_34

Tabel 4.2 Kebutuhan fungsional (lanjutan)

No.	Use Case	Aktor	Spesifikasi	Kode
35	Membarui Status Buat Akta Wasiat	Pegawai	Sistem harus dapat membarui status akta wasiat.	EPN_F_35
36	Mengonfirmasi Status Buat Akta Wasiat Selesai	Pegawai	Sistem harus dapat mengonfirmasi akta wasiat selesai.	EPN_F_36
37	Melihat Pendaftaran SK Wasiat	Pemilik, Pegawai	Sistem harus dapat menampilkan data pendaftaran SK wasiat.	EPN_F_37
38	Mengunggah Berkas SK Wasiat	Pemilik	Sistem harus dapat mengunggah berkas wasiat.	EPN_F_38
39	Melihat Rincian Berkas Klien Pada Pendaftaran SK Wasiat	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada pendaftaran SK wasiat.	EPN_F_39
40	Melihat Buat Akta Fidusia	Pemilik, Pegawai	Sistem harus dapat melihat buat akta fidusia.	EPN_F_40
41	Menambah Buat Akta Fidusia	Pegawai	Sistem harus dapat menambahkan buat akta fidusia.	EPN_F_41
42	Melihat Rincian Berkas Klien Pada Buat Akta Fidusia	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada pesan nama fidusia.	EPN_F_42
43	Memverifikasi Buat Akta Fidusia	Pemilik	Sistem harus dapat memverifikasi buat akta fidusia.	EPN_F_43
44	Menolak Buat Akta Fidusia	Pemilik	Sistem harus dapat menolak buat akta fidusia	EPN_F_44
45	Mengubah buat akta Fidusia	Pegawai	Sistem harus dapat mengubah buat akta fidusia.	EPN_F_45
46	Membarui Status Buat Akta Fidusia	Pegawai	Sistem harus dapat membarui status buat akta fidusia.	EPN_F_46

Tabel 4.2 Kebutuhan fungsional (lanjutan)

No.	Use Case	Aktor	Spesifikasi	Kode
47	Mengonfirmasi Status Buat Akta Fidusia Selesai	Pegawai	Sistem harus dapat mengonfirmasi akta wasiat selesai.	EPN_F_47
48	Melihat Pendaftaran SK Fidusia	Pemilik, Pegawai	Sistem harus dapat menampilkan data pendaftaran SK fidusia.	EPN_F_48
49	Mengunggah Berkas SK Fidusia	Pemilik	Sistem harus dapat mengunggah berkas fidusia.	EPN_F_49
50	Melihat Rincian Berkas Klien Pada Pendaftaran SK Fidusia	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada pendaftaran SK fidusia.	EPN_F_50
51	Melihat Rincian Berkas Klien Pada Protokol Notaris	Pemilik, Pegawai	Sistem harus dapat menampilkan data rincian berkas klien pada protokol notaris.	EPN_F_51
52	Mengubah Protokol Notaris	Pemilik	Sistem harus dapat Mengubah protokol notaris.	EPN_F_52
53	Menambah protokol notaris	Pemilik	Sistem harus dapat menambah protokol notaris.	EPN_F_53
54	Melihat Protokol Notaris	Pemilik, Pegawai	Sistem harus dapat melihat protokol notaris.	EPN_F_54
55	Melihat <i>Dashboard</i>	Pemilik, Pegawai	Sistem harus dapat melihat <i>Dashboard</i> .	EPN_F_55
56	Melihat Monitoring Kerja	Pemilik	Sistem harus dapat monitoring kerja.	EPN_F_56
57	Melihat <i>Log</i> Aktivitas Pegawai	Pemilik	Sistem harus dapat <i>log</i> aktivitas pegawai.	EPN_F_57
58	Melihat Notifikasi <i>Deadline</i>	Pemilik, Pegawai	Sistem harus dapat melihat notifikasi <i>deadline</i> .	EPN_F_58

Tabel 4.2 Kebutuhan fungsional (lanjutan)

No.	Use Case	Aktor	Spesifikasi	Kode
59	Melihat Notifikasi Umur Kertas	Pemilik, Pegawai	Sistem harus dapat melihat notifikasi umur kertas.	EPN_F_59
60	Menambah Informasi	Pemilik	Sistem harus dapat menambahkan informasi.	EPN_F_60
61	Mengubah Informasi	Pemilik	Sistem harus dapat mengubah informasi.	EPN_F_61
62	Menghapus Informasi	Pemilik	Sistem harus dapat menghapus informasi.	EPN_F_62

4.4 Kebutuhan Non-Fungsional

Kumpulan kebutuhan yang bersinggungan terhadap batasan dan kualitas sistem merupakan kebutuhan non-fungsional. Pemaparan kebutuhan yang berhubungan dengan karakteristik sistem, yakni *performance*, *availability*, *usability* akan disediakan oleh kebutuhan non-fungsional. Sistem pembukuan dan administrasi protokol notaris memiliki kebutuhan non-fungsional yang harus ada pada perangkat lunak ini. Kebutuhan non-fungsional dari sistem ini diutarakan pada Tabel 4.3.

Tabel 4.3 Kebutuhan non-fungsional

No	Parameter	Deskripsi Kebutuhan	Kode
1	<i>Usability</i>	<i>Usability</i> , yaitu kemudahan ketika memanfaatkan sistem yang nantinya digunakan oleh masing-masing pengguna.	EPN_NF_1

4.4.1 Diagram Use Case

Behaviour atas aktor kepada sistem dan visualisasi kemampuan yang dapat dilakukan oleh aktor terhadap sistem divisualisasikan oleh diagram *use case*. Selain itu, diagram *use case* juga memvisualisasikan hubungan antara aktor dengan eksternal sistem. Visualisasi fungsionalitas sistem yang dibuat memiliki sasaran agar dapat melihat dan mengerti terhadap visualisasi fungsionalitas sistem yang akan dibuat. Sistem pembukuan dan administrasi protokol notaris ini memiliki tiga aktor, yakni pengguna, pemilik, dan pegawai. Selain itu, terdapat 62 kebutuhan fungsional. Susunan *behaviour* dari masing-masing aktor yang telah divisualisasikan ditunjukkan dalam Gambar 4.1.

4.5 Use Case Scenario

Plot kegiatan yang terjadi antara pengguna yang menjalankan sistem didefinisikan di skenario *use case*. Selain itu, *use case* yang telah dibuat akan mengelompokkan plot kegiatan yang terjadi. Aktor, objektif, kondisi awal, aliran utama, aliran alternatif, dan kondisi akhir akan mengisi tabel skenario *use case*. Skenario *use case* ditunjukkan antara tabel 4.4 dan 4.64.

Tabel 4.4 Use case scenario login

<i>Login</i>	
Aktor	Pengguna
Objektif	Sistem mengizinkan <i>user</i> mengakses sebagai pemilik atau pegawai.
Kondisi awal	Sistem mempresentasikan halaman <i>login</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Pengguna meng-<i>input</i> <i>username</i> dan <i>password</i> dan menekan tombol <i>login</i>. 2. Halaman daftar akan tampil ketika sistem sudah menverifikasi status input dari pengguna. 3. Sistem mempresentasikan <i>dashboard</i> pengguna.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Jika <i>username</i> atau <i>password</i> salah sistem mencetak pesan "<i>Username</i> atau <i>password</i> anda salah". 2. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem mempresentasikan halaman <i>dashboard</i> dari pengguna.

Tabel 4.5 Use case scenario logout

<i>Logout</i>	
Aktor	Pemilik dan pegawai
Objektif	Untuk keluar dari sistem sebagai pengguna.
Kondisi awal	<ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam sistem sebagai pemilik atau pengguna dan berada pada halaman <i>Dashboard</i> dari pengguna.

Tabel 4.5 Use case scenario logout (lanjutan)

<i>Logout</i>	
Aliran utama	<ol style="list-style-type: none"> 2. Aktor memilih tombol nama. 3. Aktor memilih tombol logout. 4. Sistem mengeluarkan aktor dari sistem sebagai pengguna dan menampilkan halaman <i>login</i>.
Aliran alternatif:	-
Kondisi akhir	Sistem mempresentasikan halaman <i>login</i> .

Tabel 4.6 Use case scenario melihat akun pegawai

Melihat Akun	
Aktor	Pemilik
Objektif	Untuk melihat akun yang telah terdaftar.
Kondisi awal	Aktor telah <i>login</i> ke dalam sistem.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor menekan menu "Akun Pegawai". 2. Sistem merepresentasikan halaman data <i>username</i>, nama lengkap, <i>password</i>, dan status aktif.
Aliran alternatif:	-
Kondisi akhir	Sistem mempresentasikan halaman data akun pegawai yang terdapat data <i>username</i> , nama lengkap, <i>password</i> , dan status aktif.

Tabel 4.7 Use case scenario menambah akun pegawai

Menambah Akun	
Aktor	Pemilik
Objektif	Untuk menambahkan akun pegawai.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan <i>form</i> tambah data akun.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah akun yang berisi data <i>username</i>, nama lengkap,

Tabel 4.7 Use case scenario menambah akun pegawai (lanjutan)

Menambah Akun	
Aliran utama	<ol style="list-style-type: none"> 1. <i>Password, re-type password</i>, hak akses, dan status aktif. 2. Aktor memilih tombol “Tambah”. 3. Aktor mengisi <i>form</i> tambah akun yang berisi data <i>username</i>, nama lengkap, <i>password, re-type password</i>, hak akses, dan status aktif. 4. Aktor memilih tombol “Tambah”. 5. Sistem mengecek <i>username</i> telah digunakan sebelumnya atau belum, jika belum maka sistem menyimpan masukkan aktor dan mencetak pesan “Pendaftaran berhasil”.
Aliran alternatif	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan. 2. Jika <i>username</i> telah digunakan sebelumnya, sistem mencetak pesan “<i>username</i> telah digunakan”.
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan “Pendaftaran berhasil” dan mempresentasikan halaman data akun yang telah diperbarui.

Tabel 4.8 Use case scenario mengubah akun pegawai

Mengubah akun pegawai	
Aktor	Pemilik
Objektif	Untuk mengubah akun pegawai.
Kondisi awal	Aktor telah login, sistem mempresentasikan <i>form</i> data akun, dan aktor telah memilih akun yang ingin diubah.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> edit berupa data <i>username</i>, nama lengkap, <i>password</i>, hak akses, dan status aktif. 2. Aktor kemudian memilih tombol “Ubah”.



Tabel 4.8 Use case scenario mengubah akun pegawai (lanjutan)

Mengubah akun pegawai	
Aliran utama	3. Sistem menyimpan masukkan aktor dan mencetak pesan “Akun berhasil diubah”, kemudian mempresentasikan halaman data akun yang telah diperbarui.
Aliran alternatif	1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan mempresentasikan peringatan.
Kondisi akhir	Sistem menyimpan masukkan aktor dan mencetak pesan “Akun berhasil diubah”, kemudian menampilkan halaman data akun yang telah diperbarui.

Tabel 4.9 Use case scenario menghapus akun pegawai

Menghapus Akun	
Aktor	Pemilik
Objektif	Untuk menghapus akun pegawai.
Kondisi awal	Aktor telah login, sistem mempresentasikan <i>form</i> data akun, dan aktor telah memilih akun yang ingin dihapus.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih akun yang ingin dihapus. 2. Aktor memilih tombol hapus. 3. Sistem menampilkan pesan. 4. Sistem menampilkan peringatan konfirmasi. 5. Aktor memilih tombol “Ya, Hapus!”. 6. Sistem menghapus akun dari <i>database</i> dan mempresentasikan halaman data akun yang telah diperbarui.
Aliran alternatif:	1. Apabila aktor menekan tombol “Cancel” sebagai respon peringatan konfirmasi maka akun pegawai terpilih batal di hapus.

Tabel 4.2 Use case scenario menghapus akun (lanjutan)

Menghapus Akun	
Kondisi akhir	Sistem mempresentasikan halaman data akun yang telah diperbarui.

Tabel 4.3 Use case scenario menambah klien

Menambah Klien	
Aktor	Pegawai
Objektif	Untuk menambahkan klien.
Kondisi awal	Aktor telah login dan sistem mempresentasikan <i>form</i> tambah data klien.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah klien yang data nomor KTP, nama lengkap, tanggal lahir, nomor HP, dan instansi. 2. Aktor kemudian memilih tombol "Tambah". 3. Sistem mengecek <i>unique</i> id telah digunakan sebelumnya atau belum, jika belum maka sistem menyimpan masukkan aktor dan mencetak pesan "Pendaftaran berhasil".
Aliran alternatif:	<ol style="list-style-type: none"> 1. Jika nomor KTP telah digunakan sebelumnya, sistem mencetak pesan "Nomor KTP telah terdaftar". 2. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Pendaftaran berhasil" dan menampilkan halaman data akun yang telah di diperbarui.

Tabel 4.4 Use case scenario melihat klien

Melihat Klien	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat klien yang telah terdaftar.

Tabel 4.11 Use case scenario melihat klien (lanjutan)

Melihat Klien	
Kondisi awal	Aktor telah <i>login</i> ke dalam sistem.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor menekan menu Klien. 2. Sistem mempresentasikan halaman data data nomor KTP, nama lengkap, tanggal lahir, nomor HP, dan instansi.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data klien yang terdapat data data nomor KTP, nama lengkap, tanggal lahir, nomor HP, dan instansi.

Tabel 4.12 Use case scenario mengubah klien

Mengubah Klien	
<i>Actor</i>	Pemilik dan pegawai
<i>Objective</i>	Untuk mengubah klien.
<i>Pre-Condition</i>	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> mengubah klien.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> ubah klien berupa data nomor KTP, nama lengkap, tanggal lahir, nomor HP, dan instansi. 2. Aktor memilih tombol "Ubah". 3. Sistem menyimpan masukkan actor dan menampilkan pesan "Klien berhasil diubah", kemudian menampilkan halaman data akun yang telah diperbarui.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
<i>Post-Condition</i>	Sistem menyimpan masukkan aktor dan mencetak pesan "Klien berhasil diubah", kemudian mempresentasikan halaman data klien yang telah diperbarui.

Tabel 4.13 Use case scenario melihat riwayat klien

Melihat Riwayat Klien	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat riwayat order klien.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman daftar riwayat klien.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu Klien. 2. Aktor memilih klien dan memilih tombol riwayat klien.
Aliran alternatif	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman riwayat klien yang terdapat rincian riwayat order klien.

Tabel 4.14 Use case scenario melihat pesan nama organisasi

Melihat Pesan Nama Organisasi	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat pesan nama organisasi.
Kondisi awal	Aktor telah <i>login</i> dan sistem menampilkan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu pesan nama. 3. Sistem mempresentasikan halaman data pesan nama yang terdapat data klien, tipe organisasi, nama organisasi, tanggal mulai, tanggal selesai, petugas, instansi, status, dan keterangan tambahan.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data pesan nama yang terdapat data klien, tipe organisasi, nama organisasi, tanggal mulai, tanggal selesai, petugas, instansi, status, dan keterangan tambahan

Tabel 4.15 Use case scenario menambah pesan nama organisasi

Menambah Pesan Nama Organisasi	
Aktor	Pegawai
Objektif	Untuk menambahkan pesan nama organisasi organisasi.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> tambah pesan nama.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah pesan nama organisasi yang berisi data klien, tipe organisasi, nama organisasi, tanggal mulai, tanggal selesai, keterangan tambahan, dan berkas unggahan. 2. Aktor memilih tombol “Tambah”. 3. Sistem menyimpan masukan aktor dan mencetak pesan “Order berhasil ditambah”.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila <i>file</i> yang diunggah bukan ekstensi .pdf. sistem mencetak pesan “Berkas gagal diunggah”. 2. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem mencetak pesan “Order berhasil ditambah” dan menampilkan halaman data order yang telah diperbarui.

Tabel 4.16 Use case scenario melihat rincian berkas klien pada pesan nama organisasi

Melihat Rincian Berkas Klien pada Pesan Nama Organisasi	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada pesan nama organisasi.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu pesan nama organisasi.

Tabel 4.16 Use case scenario melihat rincian berkas klien pada pesan nama organisasi (lanjutan)

Melihat Rincian Berkas Klien pada Pesan Nama Organisasi	
Aliran utama	<ol style="list-style-type: none"> 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada pesan nama organisasi.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada pesan nama organisasi.

Tabel 4.17 Use case scenario mengubah pesan nama organisasi

Mengubah Pesan Nama Organisasi	
Aktor	Pegawai
Objektif	Untuk mengubah pesan nama organisasi.
Kondisi awal	Aktor telah berhasil <i>login</i> , sistem mempresentasikan <i>form</i> mengubah pesan nama organisasi, dan aktor telah memilih pesan nama organisasi yang ingin diubah.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> ubah pesan nama organisasi yang berisi data klien, tipe organisasi, nama organisasi, tanggal mulai, tanggal selesai, dan keterangan tambahan. 2. Aktor memilih tombol "Ubah". 3. Sistem menyimpan masukan aktor dan mencetak pesan "Order berhasil diubah", kemudian menampilkan data order yang telah diperbarui.
Aliran alternatif:	1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukan actor dan mencetak pesan "Order berhasil diubah" dan mempresentasikan halaman data order yang telah diperbarui.



Tabel 4.18 Use case scenario menolak pesan nama organisasi

Menolak Pesan Nama Organisasi	
Aktor	Pemilik
Objektif	Untuk menolak pesan nama organisasi.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> tolak pesan nama organisasi.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi form tolak pesan nama organisasi. 2. Aktor memilih tombol "Tolak". 3. Sistem menyimpan masukan aktor dan mempresentasikan halaman data pesan nama organisasi.
Aliran alternatif:	1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan "Order berhasil ditolak" dan mempresentasikan halaman data pesan nama organisasi.

Tabel 4.19 Use case scenario menghapus pesan nama organisasi

Menghapus Pesan Nama Organisasi	
Aktor	Pemilik
Objektif	Untuk menghapus pesan nama organisasi.
Kondisi awal	Aktor telah login, sistem mempresentasikan <i>form</i> data pesan nama organisasi, dan aktor telah memilih pesan nama organisasi yang ingin dihapus.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih akun yang ingin dihapus. 2. Aktor memilih tombol hapus. 3. Sistem menghapus pesan nama organisasi dari <i>database</i> dan mempresentasikan halaman data pesan nama yang telah diperbarui.
Aliran alternatif:	-



Tabel 4.19 Use case scenario menghapus pesan nama organisasi (lanjutan)

Menghapus Pesan Nama Organisasi	
Kondisi akhir	Sistem mempresentasikan halaman data akun yang telah diperbarui.

Tabel 4.20 Use case scenario memverifikasi pesan nama organisasi

Memverifikasi Pesan Nama Organisasi.	
Aktor	Pemilik
Objektif	Untuk memverifikasi pesan nama organisasi.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> data pesan nama organisasi.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data order kemudian memilih tombol verifikasi pesan nama organisasi. 2. Sistem menyimpan masukkan aktor dan mencetak pesan "Order berhasil terverifikasi".
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Order berhasil terverifikasi" dan menampilkan halaman

Tabel 4.21 Use case scenario melihat buat akta organisasi

Melihat Buat Akta Organisasi	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat buat akta organisasi.
Kondisi awal	Aktor telah <i>login</i> dan sistem menampilkan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu buat akta.

Tabel 4.21 Use case scenario melihat buat akta organisasi (lanjutan)

Melihat Buat Akta Organisasi	
Aliran utama	3. Sistem mempresentasikan halaman data buat akta yang terdapat data klien, tipe organisasi, nama organisasi, tanggal mulai, tanggal selesai, petugas, instansi, status, dan keterangan tambahan.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data buat akta yang terdapat data klien, tipe organisasi, nama organisasi, tanggal mulai, tanggal selesai, petugas, instansi, keterangan tambahan, dan status.

Tabel 4.22 Use case scenario melihat rincian berkas klien pada buat akta organisasi

Melihat Rincian Berkas Klien pada Buat Akta Organisasi	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada buat akta organisasi.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu buat akta organisasi. 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada buat akta organisasi.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada buat akta organisasi.

Tabel 4.23 Use case scenario membarui status buat akta organisasi

Membarui Status Buat Akta Organisasi



Tabel 4.23 Use case scenario membarui status buat akta organisasi (lanjutan)

Membarui Status Buat Akta Organisasi	
Aktor	Pegawai
Objektif	Untuk membarui status buat akta organisasi.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> buat akta organisasi.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data buat akta organisasi kemudian memilih tombol <i>update</i> dan memilih status yang diinginkan. 2. Sistem menyimpan masukan aktor dan menampilkan halaman data buat akta organisasi.
Aliran utama:	-
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan "Status order telah diperbarui" dan mempresentasikan halaman data buat akta organisasi.

Tabel 4.24 Use case scenario mengonfirmasi status buat akta organisasi selesai

Mengonfirmasi Status Buat Akta Organisasi Selesai	
Aktor	Pegawai
Objektif	Untuk mengonfirmasi status buat akta organisasi selesai.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> buat akta organisasi.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data buat akta organisasi kemudian memilih tombol konfirmasi selesai. 2. Sistem menyimpan masukan aktor dan mempresentasikan halaman data buat akta organisasi.
Aliran alternatif:	-



Tabel 4.24 Use case scenario mengonfirmasi status buat akta organisasi selesai (lanjutan)

Mengonfirmasi Status Buat Akta Organisasi Selesai	
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Status order telah diperbarui" dan mempresentasikan halaman data buat akta organisasi.

Tabel 4.25 Use case scenario melihat pendaftaran SK organisasi

Melihat Pendaftaran SK Organisasi	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat pendaftaran SK organisasi.
Kondisi awal	Aktor sudah mengakses fitur <i>login</i> dan halaman <i>dashboard</i> direpresentasikan ke dalam sistem.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu pendaftaran SK. 3. Sistem mempresentasikan halaman data pendaftaran SK yang terdapat data klien, nama organisasi, tanggal selesai, dan keterangan tambahan.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data pendaftaran SK yang terdapat data klien, nama organisasi, tanggal selesai, dan keterangan tambahan.

Tabel 4.26 Use case scenario mengunggah berkas SK organisasi

Mengunggah Berkas SK Organisasi	
Aktor	Pemilik
Objektif	Untuk mengunggah berkas SK organisasi.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .

Tabel 4.26 Use case scenario mengunggah berkas SK organisasi (lanjutan)

Mengunggah Berkas SK Organisasi	
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu pendaftaran SK. 3. Aktor memilih data berkas SK kemudian memilih tombol unggah berkas SK. 4. Aktor memilih tombol <i>choose file</i>. 5. Aktor memilih dan mengunggah berkas. 6. Aktor memilih tombol <i>open</i>. 7. Aktor menekan tombol <i>upload</i>. 8. Sistem menyimpan masukan aktor dan mencetak pesan "Berkas berhasil diunggah".
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila <i>file</i> yang diunggah bukan ekstensi <i>.pdf</i>. sistem mencetak pesan "Berkas gagal diunggah".
Kondisi akhir	Sistem menyimpan masukan aktor dan mencetak pesan "Berkas berhasil diunggah".

Tabel 4.27 Use case scenario melihat rincian berkas klien pada pendaftaran SK organisasi

Melihat Rincian Berkas Klien pada Pendaftaran SK Organisasi	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada pendaftaran sk organisasi
Kondisi awal	Aktor telah <i>login</i> dan sistem menampilkan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu organisasi. 2. Aktor memilih sub-menu buat pendaftaran SK. 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem menampilkan rincian berkas klien pada pendaftaran SK organisasi
Aliran alternatif:	-

Tabel 4.27 Use case scenario melihat rincian berkas klien pada pendaftaran SK organisasi (lanjutan)

Melihat Rincian Berkas Klien pada Pendaftaran SK Organisasi	
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada pendaftaran SK organisasi.

Tabel 4.28 Use case scenario melihat cek nama wasiat

Melihat Cek Nama Wasiat	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat cek nama wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu cek nama. 3. Sistem mempresentasikan halaman data cek nama yang terdapat data klien, nama pewasiat, tanggal mulai, tanggal selesai, petugas, instansi, status, dan keterangan tambahan.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data cek nama yang terdapat data klien, nama pewasiat, tanggal mulai, tanggal selesai, petugas, instansi, status, dan keterangan tambahan.

Tabel 4.29 Use case scenario menambah cek nama wasiat

Menambah Cek Nama Wasiat	
Aktor	Pegawai
Objektif	Untuk menambahkan cek nama wasiat.
Kondisi awal	Aktor telah akses fitur <i>login</i> dan <i>form</i> tambah cek nama wasiat telah direpresentasikan.

Tabel 4.29 Use case scenario menambah cek nama wasiat (lanjutan)

Menambah Cek Nama Wasiat	
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah cek nama wasiat yang berisi data klien, nama pewasiat, tanggal mulai, tanggal selesai, keterangan tambahan, dan berkas unggahan. 2. Aktor memilih tombol "Tambah". 3. Sistem menyimpan masukan aktor dan mencetak pesan "Order berhasil ditambah".
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila <i>file</i> yang diunggah bukan ekstensi .pdf sistem mencetak pesan "Berkas gagal diunggah". 2. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem mencetak pesan "Order berhasil ditambah" dan mempresentasikan halaman data cek nama wasiat yang telah diperbarui.

Tabel 4.30 Use case scenario melihat rincian berkas klien pada cek nama wasiat

Melihat Rincian Berkas Klien pada Cek Nama Wasiat	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada cek nama wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu cek nama. 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada cek nama wasiat.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada cek nama wasiat.



Tabel 4.31 Use case scenario memverifikasi cek nama wasiat

Memverifikasi Cek Nama Wasiat	
Aktor	Pemilik
Objektif	Untuk memverifikasi pesan nama wasiat.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> data cek nama wasiat.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data order kemudian memilih tombol verifikasi cek nama wasiat. 2. Sistem menyimpan masukkan aktor dan mencetak pesan "Order berhasil terverifikasi".
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Order berhasil diverifikasi" dan menampilkan halaman cek nama wasiat.

Tabel 4.32 Use case scenario menolak cek nama wasiat

Menolak Cek Nama Wasiat	
Aktor	Pemilik.
Objektif	Untuk menolak cek nama wasiat.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> tolak cek nama wasiat.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi form tolak pesan nama wasiat. 2. Aktor memilih tombol "Tolak". 3. Sistem menyimpan masukan aktor dan mempresentasikan halaman data cek nama wasiat.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.

Tabel 4.5 Use case scenario menolak cek nama wasiat (lanjutan)

Menolak Cek Nama Wasiat	
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan “Order berhasil ditolak” dan mempresentasikan halaman data cek nama wasiat.

Tabel 4.6 Use case scenario mengubah cek nama wasiat

Mengubah Cek Nama Wasiat	
Aktor	Pegawai
Objektif	Untuk mengubah cek nama wasiat.
Kondisi awal	Aktor telah berhasil <i>login</i> , sistem mempresentasikan <i>form</i> mengubah cek nama wasiat, dan aktor telah mempresentasikan cek nama wasiat yang ingin diubah.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> ubah cek nama yang berisi data klien, nama pewasiat, tanggal mulai, tanggal selesai, dan keterangan tambahan. 2. Aktor memilih tombol “Ubah”. 3. Sistem menyimpan masukan aktor dan mencetak pesan “Order berhasil diubah”, kemudian mempresentasikan data cek nama wasiat yang telah diperbarui.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukan actor dan mencetak pesan “Order berhasil diubah” dan mempresentasikan halaman data cek nama wasiat yang telah diperbarui.

Tabel 4.7 Use case scenario memverifikasi status pewasiat

Memverifikasi Status Pewasiat	
Aktor	Pemilik



Tabel 4.34 Use case scenario memverifikasi status pewasiat (lanjutan)

Memverifikasi Status Pewasiat	
Objektif	Untuk memverifikasi status pewasiat.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> data cek nama wasiat.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data order kemudian memilih tombol verifikasi cek nama wasiat. 2. Sistem menyimpan masukkan aktor dan mencetak pesan "Pewasiat berhasil terverifikasi".
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Pewasiat berhasil terverifikasi" dan mempresentasikan halaman data cek nama wasiat yang telah di diperbarui.

Tabel 4.35 Use case scenario melihat buat akta wasiat

Melihat Buat Akta Wasiat	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat cek nama wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu buat akta. 3. Sistem mempresentasikan halaman data buat akta yang terdapat data klien, nama pewasiat, tanggal mulai, tanggal selesai, petugas, keterangan tambahan, dan status.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data buat akta yang terdapat data klien, nama pewasiat, tanggal mulai, tanggal selesai, petugas, keterangan tambahan, dan status.

Tabel 4.36 Use case scenario melihat rincian berkas klien pada buat akta wasiat

Melihat Rincian Berkas Klien pada Buat Akta Wasiat	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada buat akta wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu buat akta wasiat. 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada buat akta wasiat.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada buat akta wasiat.

Tabel 4.37 Use case scenario membarui status buat akta wasiat

Membarui Status Buat Akta Wasiat	
Aktor	Pegawai
Objektif	Untuk membarui status buat akta wasiat.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> buat akta wasiat.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data buat akta wasiat kemudian memilih tombol <i>update</i> dan memilih status yang diinginkan. 2. Sistem menyimpan masukan aktor dan mempresentasikan halaman data buat akta wasiat.
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan "Status order telah diperbarui" dan mempresentasikan halaman data buat akta wasiat.

Tabel 4.38 Use case scenario mengonfirmasi status buat akta wasiat selesai

Mengonfirmasi Status Buat Akta Wasiat Selesai	
Aktor	Pegawai
Objektif	Untuk mengonfirmasi status buat akta wasiat selesai.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> buat akta wasiat.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data buat akta wasiat kemudian memilih tombol konfirmasi selesai 2. Sistem menyimpan masukan aktor dan mempresentasikan halaman data buat akta wasiat.
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan "Status order telah diperbarui" dan mempresentasikan halaman data buat akta wasiat.

Tabel 4.39 Use case scenario melihat pendaftaran SK wasiat

Melihat Pendaftaran SK Wasiat	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat pendaftaran SK wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem menampilkan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu pendaftaran SK. 3. Sistem mempresentasikan halaman data pendaftaran SK yang terdapat data klien, nama pewasiat, tanggal selesai, petugas dan keterangan tambahan.
Aliran alternatif:	-

Tabel 4.39 Use case scenario melihat pendaftaran SK wasiat (lanjutan)

Melihat Pendaftaran SK Wasiat	
Kondisi akhir	Sistem berhasil mempresentasikan halaman data pendaftaran SK yang terdapat data klien, nama pewasiat, tanggal selesai, petugas dan keterangan tambahan.

Tabel 4.40 Use case scenario mengunggah berkas SK wasiat

Mengunggah Berkas SK Wasiat	
Aktor	Pemilik
Objektif	Untuk mengunggah berkas SK wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu pendaftaran SK. 3. Aktor memilih data SK kemudian memilih tombol unggah berkas SK wasiat. 4. Aktor memilih tombol <i>choose file</i>. 5. Aktor memilih dan mengunggah berkas. 6. Aktor memilih tombol <i>open</i> 7. Aktor memilih tombol <i>upload</i>. 8. Sistem menyimpan masukan aktor dan mencetak pesan "Berkas berhasil diunggah".
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila <i>file</i> yang diunggah bukan ekstensi <i>.pdf</i>. sistem menampilkan pesan "Berkas gagal diunggah".
Kondisi akhir	Sistem menyimpan masukan aktor dan mencetak pesan "Berkas berhasil diunggah".

Tabel 4.41 Use case scenario melihat rincian berkas klien pada pendaftaran SK wasiat

Melihat Rincian Berkas Klien pada Pendaftaran SK Wasiat

Tabel 4.41 Use case scenario melihat rincian berkas klien pada pendaftaran SK wasiat (lanjutan)

Melihat Rincian Berkas Klien pada Pendaftaran SK Wasiat	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada pendaftaran SK wasiat.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu wasiat. 2. Aktor memilih sub-menu pendaftaran SK. 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada pendaftaran SK wasiat.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada pendaftaran SK wasiat.

Tabel 4.42 Use case scenario melihat buat akta fidusia

Melihat Buat Akta Fidusia	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat buat akta fidusia.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu fidusia. 2. Aktor memilih sub-menu buat akta. 3. Sistem mempresentasikan halaman data buat akta fidusia yang terdapat data klien, nama fidusia, tanggal mulai, tanggal selesai, petugas, keterangan tambahan, dan status.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data buat akta fidusia yang terdapat data klien, nama fidusia,

Tabel 4.42 Use case scenario melihat buat akta fidusia (lanjutan)

Melihat Buat Akta Fidusia	
Kondisi akhir	tanggal mulai, tanggal selesai, petugas, keterangan tambahan, dan status.

Tabel 4.43 Use case scenario menambah buat akta fidusia

Menambah Buat Akta Fidusia	
Aktor	Pegawai
Objektif	Untuk menambahkan buat akta nama organisasi.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> tambah buat akta fidusia.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah buat akta fidusia yang berisi data klien, nama fidusia, tanggal mulai, tanggal selesai, keterangan tambahan, dan berkas unggahan. 2. Aktor memilih tombol "Tambah". 3. Sistem menyimpan masukan aktor dan mencetak pesan "Order berhasil ditambah".
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila <i>file</i> yang diunggah bukan ekstensi .pdf sistem menampilkan pesan "Berkas gagal diunggah". 2. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem mencetak pesan "Order berhasil ditambah" dan mempresentasikan halaman data buat akta fidusia yang telah diperbarui.

Tabel 4.44 Use case scenario melihat rincian berkas klien pada buat akta fidusia

Melihat Rincian Berkas Klien pada Buat Akta Fidusia	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada buat akta fidusia.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu fidusia. 2. Aktor memilih sub-menu buat akta fidusia. 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada buat akta fidusia
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada buat akta fidusia.

Tabel 4.45 Use case scenario memverifikasi buat akta fidusia

Memverifikasi Buat Akta Fidusia	
Aktor	Pemilik
Objektif	Untuk memverifikasi buat akta fidusia.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> data buat akta fidusia.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data order kemudian memilih tombol verifikasi buat akta fidusia. 2. Sistem menyimpan masukkan aktor dan mencetak pesan "Order berhasil terverifikasi".
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Order berhasil diverifikasi" dan mempresentasikan halaman buat akta fidusia.



Tabel 4.46 Use case scenario menolak buat akta fidusia

Menolak Buat Akta Fidusia	
Aktor	Pemilik
Objektif	Untuk menolak buat akta fidusia.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> tolak buat akta fidusia.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi form tolak buat akta fidusia. 2. Aktor memilih tombol "Tolak". 3. Sistem menyimpan masukan aktor dan mempresentasikan halaman data buat akta fidusia.
Aliran alternatif:	1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan "Order berhasil ditolak" dan mempresentasikan halaman data buat akta fidusia.

Tabel 4.47 Use case scenario mengubah buat akta fidusia

Mengubah Buat Akta Fidusia	
Aktor	Pegawai
Objektif	Untuk mengubah buat akta fidusia.
Kondisi awal	Aktor telah berhasil <i>login</i> , sistem menampilkan <i>form</i> mengubah cek nama wasiat, dan aktor telah memilih cek nama wasiat yang ingin diubah.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> ubah buat akta fidusia yang berisi data klien, nama fidusia, tanggal mulai, tanggal selesai, keterangan tambahan, dan berkas unggahan. 2. Aktor memilih tombol "Ubah". 3. Sistem menyimpan masukan aktor dan mencetak pesan "Order berhasil diubah", kemudian mempresentasikan



Tabel 4.47 Use case scenario mengubah buat akta fidusia (lanjutan)

Mengubah Buat Akta Fidusia	
Aliran utama	4. data buat akta fidusia yang telah diperbarui.
Aliran alternatif:	Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukkan aktor dan mencetak pesan “Order berhasil diubah” dan mempresentasikan halaman data buat akta fidusia yang telah diperbarui.

Tabel 4.48 Use case scenario membarui status buat akta fidusia

Membarui Status Buat Akta Fidusia	
Aktor	Pegawai
Objektif	Untuk membarui status buat akta fidusia.
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> buat akta fidusia.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data buat akta fidusia kemudian memilih tombol <i>update</i> dan memilih status yang diinginkan. 2. Sistem menyimpan masukan aktor dan mempresentasikan halaman data buat akta fidusia.
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan “Status order telah diperbarui” dan mempresentasikan halaman data buat akta fidusia.

Tabel 4.49 Use case scenario mengonfirmasi status buat akta fidusia selesai

Mengonfirmasi Status Buat Akta Fidusia Selesai	
Aktor	Pegawai
Objektif	Untuk mengonfirmasi status buat akta fidusia selesai.



Tabel 4.49 Use case scenario mengonfirmasi status buat akta fidusia selesai (lanjutan)

Mengonfirmasi Status Buat Akta Fidusia Selesai	
Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> buat akta fidusia.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih data buat akta fidusia kemudian memilih tombol konfirmasi selesai. 2. Sistem menyimpan masukan aktor dan mempresentasikan halaman data buat akta fidusia.
Aliran alternatif:	-
Kondisi akhir	Sistem menyimpan masukan aktor, mencetak pesan "Status order telah diperbarui" dan mempresentasikan halaman data buat akta fidusia.

Tabel 4.50 Use case scenario melihat pendaftaran SK fidusia

Melihat Pendaftaran SK Fidusia	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat pendaftaran SK fidusia.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu fidusia. 2. Aktor memilih sub-menu pendaftaran SK. 3. Sistem mempresentasikan halaman data pendaftaran SK yang terdapat data klien, nama fidusia, tanggal selesai, petugas dan keterangan tambahan.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data pendaftaran SK yang terdapat data klien, nama fidusia, tanggal selesai, petugas dan keterangan tambahan.

Tabel 4.51 Use case scenario mengunggah berkas SK fidusia

Mengunggah Berkas SK Fidusia	
Aktor	Pemilik
Objektif	Untuk mengunggah berkas SK fidusia.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu fidusia. 2. Aktor memilih sub-menu pendaftaran SK. 3. Aktor memilih data SK kemudian memilih tombol unggah berkas SK fidusia. 4. Aktor memilih tombol <i>choose file</i>. 5. Aktor memilih dan mengunggah berkas. 6. Aktor memilih tombol <i>open</i>. 7. Aktor memilih tombol <i>upload</i>. 8. Sistem menyimpan masukan aktor dan mencetak pesan "Berkas berhasil diunggah".
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila <i>file</i> yang diunggah bukan ekstensi .pdf. sistem mencetak pesan "Berkas gagal diunggah".
Kondisi akhir	Sistem menyimpan masukan aktor dan mencetak pesan "Berkas berhasil diunggah".

Tabel 4.52 Use case scenario melihat rincian berkas klien pada pendaftaran SK fidusia

Melihat Rincian Berkas Klien pada Pendaftaran SK Fidusia	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada pendaftaran SK fidusia.
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu fidusia. 2. Aktor memilih sub-menu pendaftaran SK.

Tabel 4.52 Use case scenario melihat rincian berkas klien pada pendaftaran SK fidusia (lanjutan)

Melihat Rincian Berkas Klien pada Pendaftaran SK Fidusia	
Aliran utama	<ol style="list-style-type: none"> 3. Aktor memilih klien dan memilih tombol rincian berkas klien. 4. Sistem mempresentasikan rincian berkas klien pada pendaftaran SK fidusia.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada pendaftaran SK fidusia.

Tabel 4.53 Use case scenario melihat rincian berkas klien pada protokol notaris

Melihat Rincian Berkas Klien pada Protokol Notaris	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat rincian berkas klien pada protokol notaris
Kondisi awal	Aktor telah <i>login</i> dan sistem mempresentasikan halaman <i>dashboard</i> .
Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih menu protokol notaris. 2. Aktor memilih klien dan memilih tombol rincian berkas klien. 3. Sistem mempresentasikan rincian berkas klien pada protokol notaris.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan rincian berkas klien pada protokol notaris.

Tabel 4.54 Use case scenario mengubah protokol notaris

Mengubah Protokol Notaris	
Aktor	Pemilik
Objektif	Untuk mengubah data protokol notaris.
Kondisi awal	Aktor telah berhasil <i>login</i> , sistem mempresentasikan <i>form</i> mengubah

Tabel 4.54 Use case scenario mengubah protokol notaris (lanjutan)

Mengubah Protokol Notaris	
Kondisi awal	protokol notaris, dan aktor telah memilih protokol notaris yang ingin diubah.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> ubah protokol notaris. 2. Memilih tombol ubah. 3. Sistem menyimpan masukan aktor dan mencetak pesan “Protokol Notaris berhasil diubah”, kemudian menampilkan halaman nomor protokol notaris, tanggal akta, data order yang telah selesai, <i>file</i> protokol notaris, dan nomor lemari.
Aliran alternatif:	1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukan aktor dan mencetak pesan “Protokol Notaris berhasil diubah”, kemudian mempresentasikan halaman data protokol notaris yang telah diperbarui.

Tabel 4.55 Use case scenario menambah protokol notaris

Menambah Protokol Notaris	
Aktor	Pemilik
Objektif	Untuk menambahkan protokol notaris.
Kondisi awal	Aktor telah berhasil <i>login</i> dan telah menyelesaikan order.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah protokol notaris yang terdapat data nama lengkap, nama organisasi, tanggal selesai, nomor protokol, nomor loker, tanggal SK, dan dokumen. 2. Sistem menyimpan masukan aktor dan mencetak pesan “Protokol Notaris berhasil ditambah”.



Tabel 4.55 Use case scenario menambah protokol notaris (lanjutan)

Menambah Protokol Notaris	
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan. 2. Jika nomor protokol telah digunakan sebelumnya, sistem mencetak pesan "Nomor protokol telah digunakan".
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan "Protokol Notaris berhasil ditambah" dan mempresentasikan halaman data protokol notaris yang telah diperbarui.

Tabel 4.56 Use case scenario melihat protokol notaris

Melihat Protokol Notaris	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat protokol notaris yang telah terdaftar.
Kondisi awal	Aktor telah <i>login</i> dan masuk ke dalam sistem.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor mengklik menu Protokol Notaris. 2. Sistem merepresentasikan halaman data protokol notaris yang terdapat nomor protokol notaris, tanggal akta, nama order, dan nomor lemari.
Aliran alternatif:	-
Kondisi akhir	Sistem berhasil mempresentasikan halaman data protokol notaris yang terdapat nomor protokol notaris, tanggal akta, nama order, dan nomor lemari.

Tabel 4.57 Use case scenario melihat dashboard

Melihat <i>Dashboard</i>	
Aktor	Pemilik dan pegawai.
Objektif	Untuk melihat halaman <i>dashboard</i> .
Kondisi awal	Aktor telah login ke dalam sistem.



Tabel 4.8 Use case scenario melihat dashboard (lanjutan)

Melihat <i>Dashboard</i>	
Aliran Utama	1. Sistem mempresentasikan <i>dashboard</i> aktor dengan data total pekerjaan, pekerjaan dalam sebulan, pekerjaan yang sedang <i>on progress</i> dan pekerjaan yang sudah selesai.
Aliran alternatif:	-
Kondisi akhir	Sistem mempresentasikan <i>dashboard</i> aktor dengan data total pekerjaan, pekerjaan dalam sebulan, pekerjaan yang sedang <i>on progress</i> dan pekerjaan yang sudah selesai.

Tabel 4.58 Use case scenario melihat monitoring kerja

Melihat Monitoring Kerja	
Aktor	Pemilik
Objektif	Untuk melihat monitoring kerja.
Kondisi awal	Aktor telah login ke dalam sistem.
Aliran Utama	1. Aktor memilih menu monitoring kerja 2. Sistem mempresentasikan monitoring kerja berupa data petugas, nama pekerjaan, tanggal masuk order, tanggal selesai order, status order yang sedang dikerjakan, dan keterangan.
Aliran alternatif:	-
Kondisi akhir	Sistem mempresentasikan monitoring kerja berupa data petugas, nama pekerjaan, tanggal masuk order, tanggal selesai order, status order yang sedang dikerjakan, dan keterangan.

Tabel 4.59 Use case scenario melihat log aktivitas pegawai

Melihat <i>Log</i> Aktivitas Pegawai	
Aktor	Pemilik
Objektif	Untuk melihat <i>log</i> aktivitas pegawai.



Tabel 4.59 Use case scenario melihat log aktivitas pegawai (lanjutan)

Melihat Log Aktivitas Pegawai	
Kondisi awal	Aktor telah login ke dalam sistem.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor memilih menu log aktivitas pegawai. 2. Sistem mempresentasikan log aktivitas pegawai berupa data username, IP, aktivitas, dan aktivitas.
Aliran alternatif:	-
Kondisi akhir	Sistem mempresentasikan monitoring kerja berupa data username, IP, aktivitas, dan aktivitas.

Tabel 4.60 Use case scenario melihat notifikasi deadline

Melihat Notifikasi <i>Deadline</i>	
Aktor	Pemilik dan pegawai.
Objektif	Untuk melihat notifikasi <i>deadline</i> order.
Kondisi awal	Aktor telah <i>login</i> .
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor memilih tombol notifikasi <i>deadline</i>. 2. Sistem mempresentasikan notifikasi <i>deadline</i> order.
Aliran alternatif:	1. Apabila tidak ada <i>deadline</i> sistem mencetak pesan "Tidak ada <i>deadline</i> ".
Kondisi akhir	Sistem berhasil mempresentasikan notifikasi <i>deadline</i> order.

Tabel 4.61 Use case scenario melihat notifikasi umur kertas

Melihat Notifikasi Umur Kertas	
Aktor	Pemilik dan pegawai
Objektif	Untuk melihat notifikasi umur kertas.
Kondisi akhir	Aktor telah <i>login</i> ke dalam sistem.

Tabel 4.61 Use case scenario melihat notifikasi umur kertas (lanjutan)

Aliran utama	<ol style="list-style-type: none"> 1. Aktor memilih tombol notifikasi umur kertas. 2. Sistem mempresentasikan notifikasi umur kertas.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila tidak ada notifikasi sistem mencetak pesan “Tidak ada info kerusakan”.
Kondisi akhir	Sistem berhasil mempresentasikan notifikasi umur kertas.

Tabel 4.62 Use case scenario menambah informasi

Menambah Informasi	
Aktor	Pemilik
Objektif	Untuk menambahkan informasi.
Kondisi awal	Aktor telah login dan sistem menampilkan <i>form</i> tambah informasi.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> tambah informasi yang terdapat judul, tipe, dan pesan. 2. Sistem menyimpan masukkan aktor dan mencetak pesan “Informasi berhasil ditambah”.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukkan aktor, mencetak pesan “Informasi berhasil ditambah” dan mempresentasikan halaman data informasi yang telah diperbarui.

Tabel 4.63 Use case scenario mengubah informasi

Mengubah Informasi	
Aktor	Pemilik
Objektif	Untuk mengubah informasi.

Tabel 4.63 Use case scenario mengubah informasi (lanjutan)

Kondisi awal	Aktor telah berhasil <i>login</i> dan sistem mempresentasikan <i>form</i> mengubah informasi.
Aliran utama	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> ubah informasi berupa data judul, tipe, dan pesan. 2. Aktor memilih tombol "Ubah". 3. Sistem menyimpan masukkan aktor dan menampilkan pesan "Informasi berhasil diubah", kemudian menampilkan halaman data informasi yang telah diperbarui.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila terdapat kekosongan pada data yang dibutuhkan sistem akan menampilkan peringatan.
Kondisi akhir	Sistem menyimpan masukkan aktor dan mencetak pesan "Informasi berhasil diubah", kemudian mempresentasikan halaman data informasi yang telah diperbarui.

Tabel 4.64 Use case scenario menghapus informasi

Menghapus Informasi	
Aktor	Pemilik
Objektif	Untuk menghapus informasi.
Kondisi awal	Aktor telah login, sistem mempresentasikan <i>form</i> data informasi, dan aktor telah memilih informasi yang ingin dihapus.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor memilih akun yang ingin dihapus. 2. Aktor memilih tombol hapus. 3. Sistem menampilkan pesan konfirmasi. 4. Aktor memilih tombol "Ya, Hapus!". 5. Sistem menghapus informasi dari <i>database</i> dan mempresentasikan halaman data informasi yang telah diperbarui.
Aliran alternatif:	<ol style="list-style-type: none"> 1. Apabila aktor menekan tombol "Cancel" sebagai respon peringatan



Tabel 4.64 Use case scenario menghapus informasi (lanjutan)

Menghapus Informasi	
Aliran alternatif:	1. konfirmasi maka akun pegawai terpilih batal di hapus.
Kondisi akhir	Sistem menampilkan halaman data informasi yang telah diperbarui.



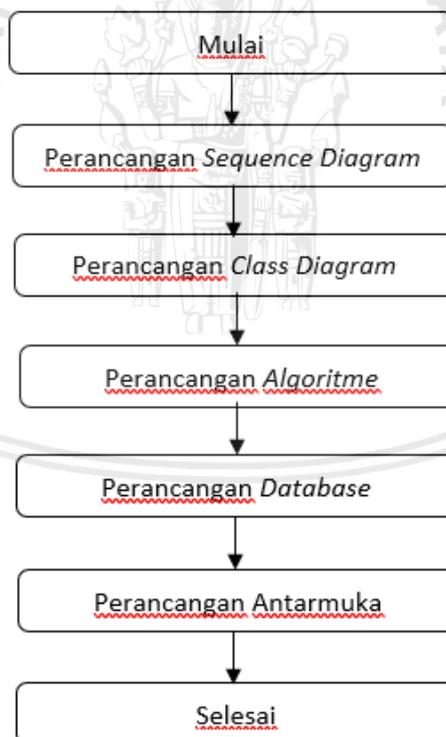
BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM

Perancangan sistem pembukuan dan administrasi protokol notaris akan dijabarkan pada bab ini. Fase perancangan dan implementasi dijalankan jika fase analisis sudah rampung. *Sequence diagram*, *class diagram*, *algoritme*, *entity relationship diagram* dan antarmuka yang dijalankan pada fase perancangan.

Sementara fase implementasi dikerjakan saat perancangan telah rampung. Implementasi dijalankan berlandaskan hasil dari bab sebelumnya. Spesifikasi terhadap sistem, batasan terhadap implementasi, implementasi basis data, dan implementasi antarmuka dijabarkan terhadap fase implementasi yang akan dilakukan.

5.1 Perancangan Sistem

Pada fase perancangan memanfaatkan *object-oriented programming (OOP)*. Perancangan *seqdiagram*, perancangan *class diagram*, perancangan *algoritme*, perancangan basis data, perancangan antarmuka dibentuk pada fase perancangan ini. Diagram alir fase perancangan sistem pembukuan dan administrasi protokol notaris ditunjukkan dalam Gambar 5.1.



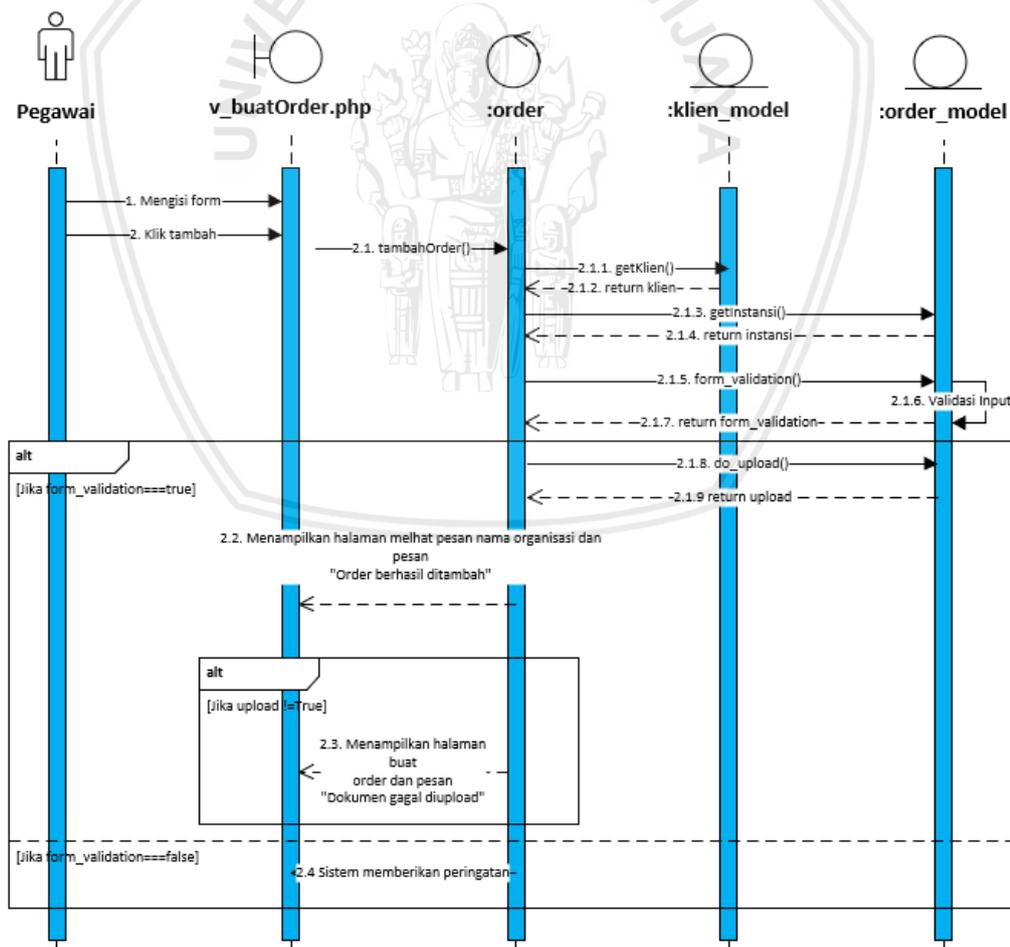
Gambar 5.1 Perancangan diagram alir sistem

5.1.1 Perancangan Sequence Diagram

Behaviour antar objek dalam susunan alir divisualisasikan oleh perancangan *sequence diagram*. *Sequence diagram* yang divisualisasikan menyelaraskan dengan *use case diagram* dan *usecase scenario*. Terdapat tiga *sequence diagram* yang dicantumkan penelitian ini.

5.1.1.1 Sequence Diagram Menambah Pesan Nama Organisasi

Sequence diagram menambah pesan nama organisasi ketika aktor mengisi form pesan nama organisasi lalu menekan tombol tambah. Selepas itu, jika validasi masukan data benar maka sistem menampilkan halaman lihat pesan nama organisasi dan menampilkan pesan order berhasil ditambah. Tetapi apabila *file* yang diunggah bukan ekstensi *.pdf* maka sistem akan menunjukkan halaman buat pesan nama organisasi dan menampilkan pesan order gagal ditambah. Kemudian apabila terdapat kekosongan pada data yang dibutuhkan sistem akan memberikan peringatan. Dalam Gambar 5.2 ditunjukkan *sequence diagram* menambah pesan nama organisasi.

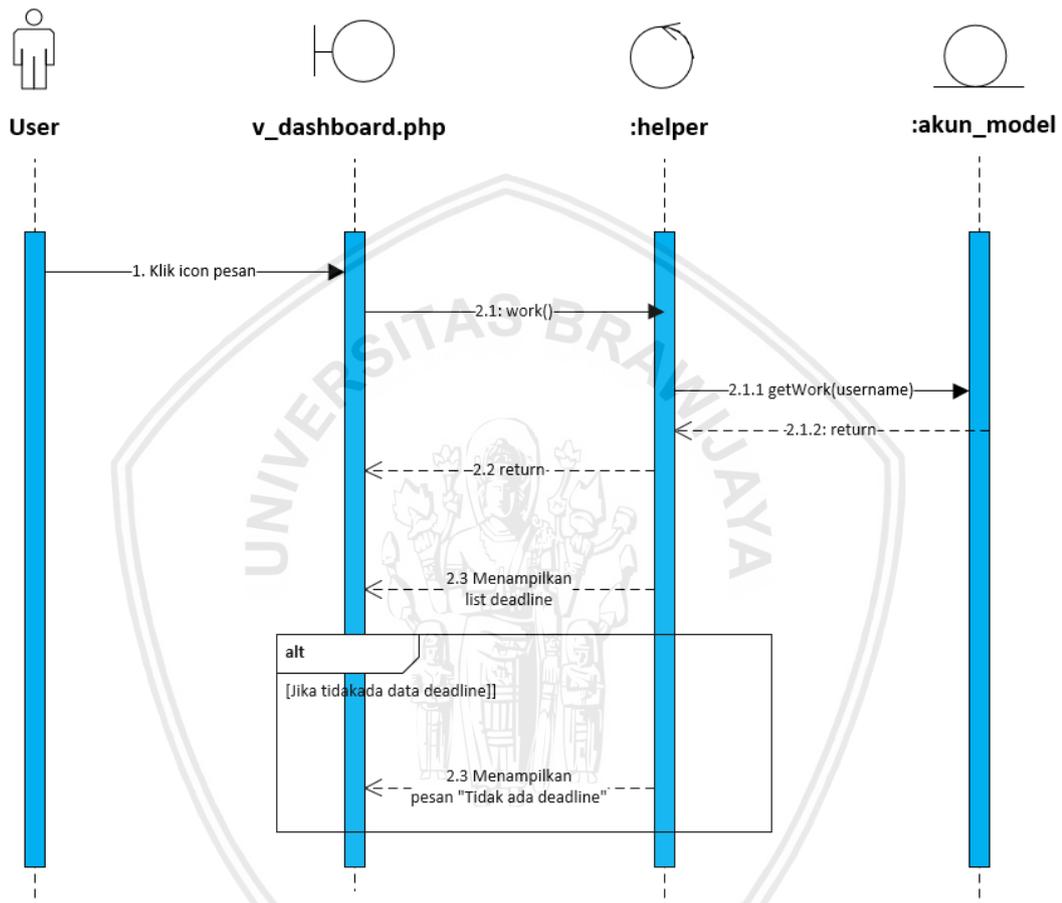


Gambar 5.2 Sequence diagram menambah pesan nama organisasi



5.1.1.2 Sequence Diagram Melihat Notifikasi Deadline

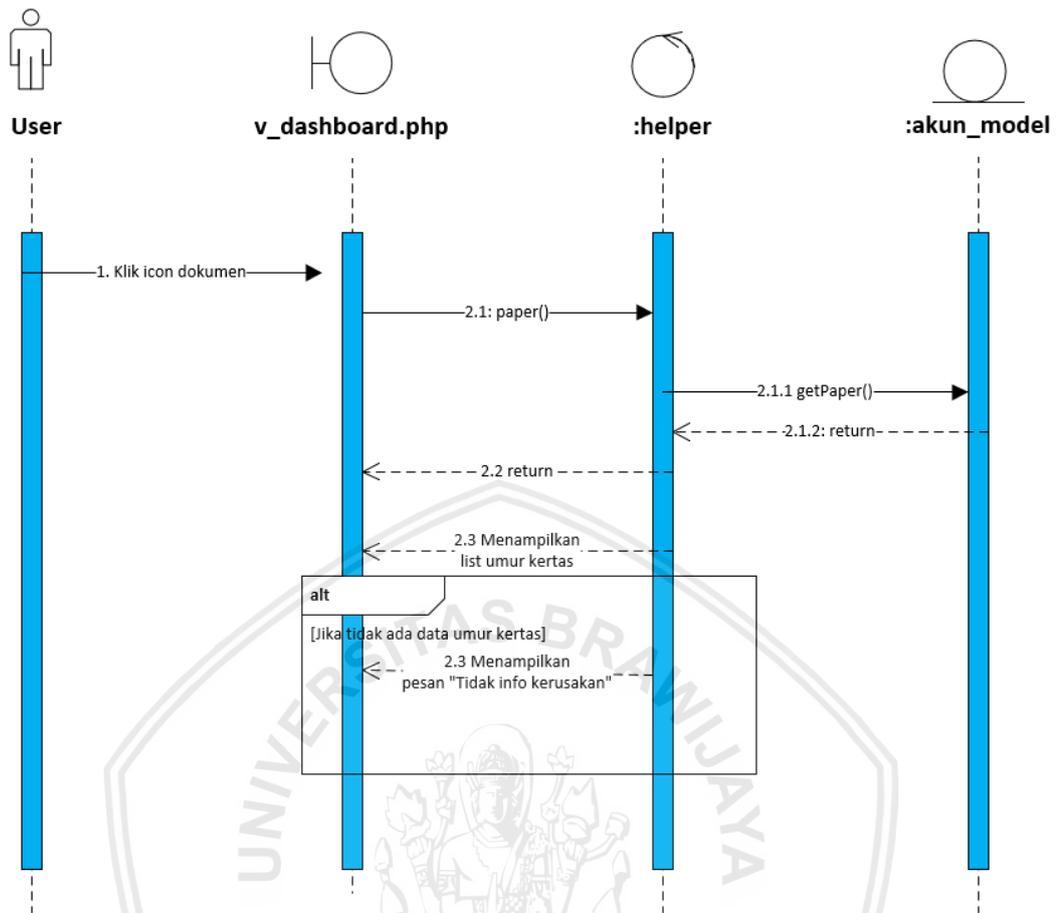
Sequence diagram melihat notifikasi deadline menjabarkan *behaviour* antar objek dalam susunan diagram alir dari proses menunjukkan notifikasi *deadline*. Prosedur akan dimulai saat aktor menentukan pilihan terhadap menu ikon notifikasi *deadline*. Lalu diteruskan prosedur pengambilan data pada basis data. Selepas itu, data ditunjukkan melalui *javascript*. Dalam Gambar 5.3 ditunjukkan *sequence diagram* melihat notifikasi *deadline*.



Gambar 5.3 Sequence diagram melihat notifikasi deadline

5.1.1.3 Sequence Diagram Melihat Notifikasi Umur Kertas

Sequence diagram melihat notifikasi umur kertas menjabarkan *behaviour* antar objek dalam susunan diagram alir dari prosedur tampilkan notifikasi umur kertas. Prosedur akan dimulai ketika aktor menentukan pilihan terhadap menu ikon notifikasi *deadline*. Lalu diteruskan prosedur pengambilan data di basis data. Kemudian, data ditampilkan melalui *javascript*. Dalam Gambar 5.4 ditunjukkan *sequence diagram* melihat notifikasi umur kertas.



Gambar 5.4 Sequence diagram melihat notifikasi umur kertas

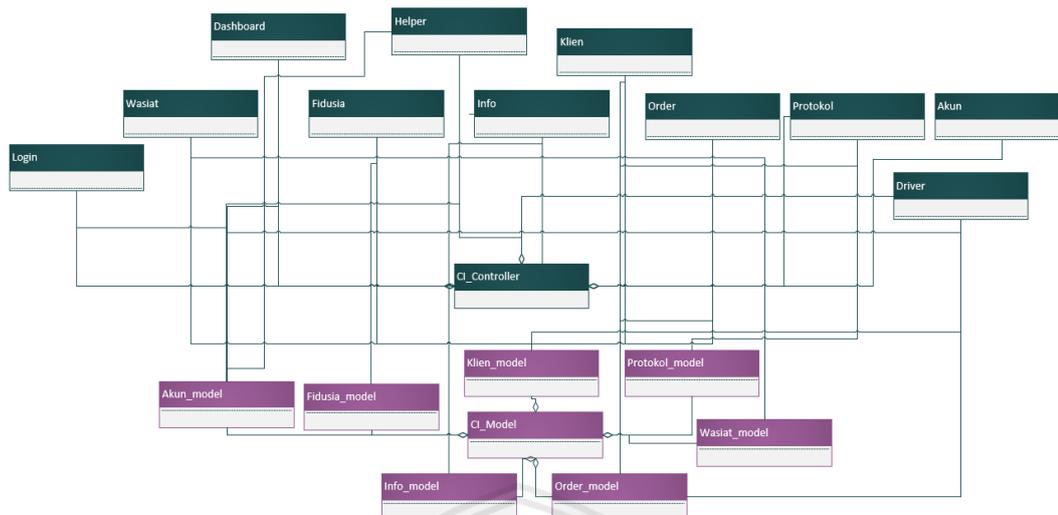
5.1.1.2 Perancangan Class Diagram

Perancangan umum dan perancangan detail diperoleh pada tahapan perancangan *class diagram*. Pada perancangan *class diagram* ini, direpresentasikan *class* ke *class* yang tercantum dalam sistem. Perancangan umum akan menjabarkan relasi-relasi antar kelas. Kemudian, penjabaran atribut dan operasi yang terdapat pada *class* akan dijelaskan perancangan detail.

5.1.1.2.1 Perancangan Umum

Pendeskripsian objek-objek yang terbentuk dengan relasi-relasi dilakukan oleh perancangan umum. Berikut merupakan perancangan umum *class diagram* atas sistem pembukuan dan administrasi protokol notaris. Perancangan *class diagram* ini menggunakan tipe MVC (*Model-View-Controller*). Tercantum kelas *controller* mempunyai kaitan asosiasi terhadap kelas *view* sedangkan kelas *view* mempunyai kaitan asosiasi terhadap kelas *controller*. Dalam Gambar 5.5 ditunjukkan perancangan umum *class diagram*.

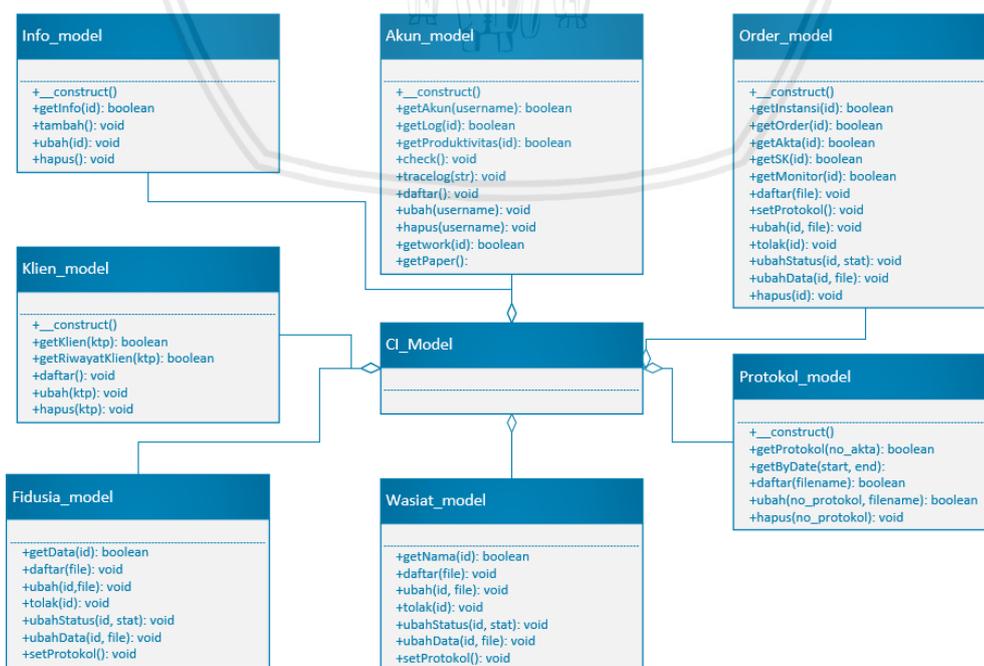




Gambar 5.5 Class diagram perancangan umum

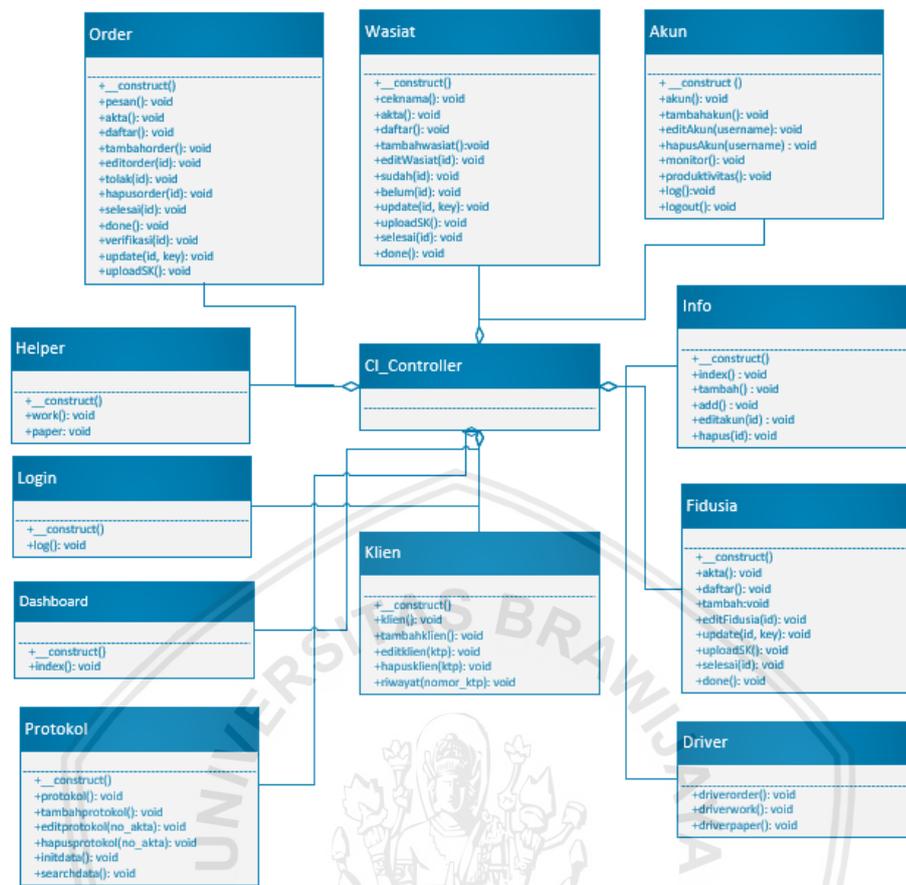
5.1.2.2 Perancangan Detail

Dua elemen terurai pada visualisasi *class diagram* perancangan detail. Bagian elemen menjabarkan kelas model yang ditunjukkan dalam Gambar 5.6. Kelas model memiliki hubungan generalisasi terhadap CI model. Kedua, Gambar 5.7 ditunjukkan kelas *controller*. Kelas *controller* memiliki hubungan generalisasi terhadap kelas CI_controller dan memiliki hubungan asosisasi terhadap kelas-kelas model. *Class diagram* di *model* termuat tujuh kelas, yakni kelas Info_model, Klien_model, Fidusia_model, Akun_model, Wasiat_model, Order_model, dan Protokol_model. Order, Helper, Login, Dashboard, Wasiat, Protokol, Klien, Akun, Info, dan Fidusia merupakan sepuluh kelas turunan pada *controller* yang dimiliki pada *class diagram*.



Gambar 5.6 Class diagram pada Model





Gambar 5.7 Class diagram pada Controller

5.1.3 Perancangan Algoritme

Prosedur yang berlaku terhadap sebagian bagian dari sub-sistem diterangkan pada perancangan algoritme. Selain itu, hal ini digunakan untuk menjalankan fungsional sistem. Dalam implementasi kode program, akan dimanfaatkan perancangan algoritme yang dibuat. Algoritme tambahOrder, algoritme work, dan algoritme paper menjadi tiga sampel dalam perancangan algoritme.

5.1.3.1 Perancangan Algoritme tambahOrder

Prosedur ketika pegawai ingin menambah order pesan nama organisasi akan dijelaskan pada algoritme fungsi tambahOrder. Apabila masukan data valid, maka data akan tersimpan, menampilkan halaman tambah akun dan menampilkan pesan pendaftaran berhasil. Namun, apabila masukan data tidak valid maka data tidak tersimpan, memperlihatkan halaman tambah akun dan menampilkan pesan pendaftaran gagal. Pada Tabel 5.1 mengutarakan algoritme tambahOrder.

Nama Kelas: Order

Nama Fungsi: tambahorder()

Algoritme:



Tabel 5.1 Algoritme fungsi tambahOrder

1	Mulai
2	Jika user tidak sama dengan pegawai
3	Kembali ke halaman order
4	Else
5	Deklarasi data log ke dalam variabel mode
6	Deklarasi data klien ke dalam variabel klien
7	Deklarasi data instansi ke dalam variabel instansi
8	Deklarasi data nama ke dalam variabel username
9	Membuat form validasi dengan aturan tertentu
10	Jika form validasi salah
11	Memuat halaman dari header
12	Memuat halaman dari nav
13	Memuat halaman dari v_buatOrder
14	Memuat halaman dari footer
15	Else
16	Mengatur upload path ke dalam /dokumen/order
17	Mengatur tipe yang diperbolehkan yaitu .pdf
18	Mengatur maksimal ukuran upload yaitu 12 mb
19	Memasukan library upload dengan aturan yang telah dibuat
20	Jika gagal menjalankan metode do_upload
21	Tampilkan display_errors
22	Else
23	Mengambil data upload
24	Memasukan ke dalam database melalui model order dengan method daftar
25	Kembali ke halaman order
26	selesai

5.1.3.2 Perancangan Algoritme work

Proses ketika pemilik dan pegawai mendapatkan notifikasi *deadline* dari pekerjaan yang sedang dikerjakan akan dijelaskan pada algoritme fungsi work. Pada Tabel 5.2 diutarakan algoritme work.

Nama Kelas: Helper

Nama Fungsi: work()

Algoritme:

Tabel 5.2 Algoritme fungsi work

1	mulai
2	jika Jika user tidak sama dengan pegawai
3	mengambil data dari model akun di method getwork tanpa parameter
4	dan disimpan di variabel data
5	else
6	mengambil data dari model akun di method getwork dengan parameter
7	username dan disimpan di variabel data
8	deklarasi variabel result
9	jika variabel data nilainya null
10	mencetak notifikasi tidak ada <i>deadline</i>
11	else
12	melakukan looping foreach memecah variabel data ke dalam variabel key
13	jika variabel key dengan index tanggal selesai lebih dari tanggal saat ini
14	deklarasi variabel datetime1 dengan nilai tanggal saat ini
15	deklarasi datetime2 dengan nilai variabel key index tanggal selesai

Tabel 5.2 Algoritme fungsi work (lanjutan)

16	deklarasi variabel deifference dengan nilai perbedaan antara datetime 2 dengan datetimel
17	jika variabel difference index days kurang dari sama dengan 7 deklarasi variabel nama dengan nilai dari variabel key index nama klien
18	menambahkan isi variabel result
19	jika variabel result nilai sama dengan kosong
20	mencetak tidak ada <i>deadline</i>
21	else
22	mencetak variabel result
23	selesai

5.1.3.3 Perancangan Algoritme paper

Proses ketika pemilik dan pegawai mendapatkan notifikasi umur kertas dari protokol notaris yang telah disimpan didalam *database* akan dijelaskan pada algoritme fungsi work. Pada Tabel 5.3 diutarakan algoritme fungsi paper.

Nama Kelas: Helper.

Nama Fungsi: paper()

Algoritme:

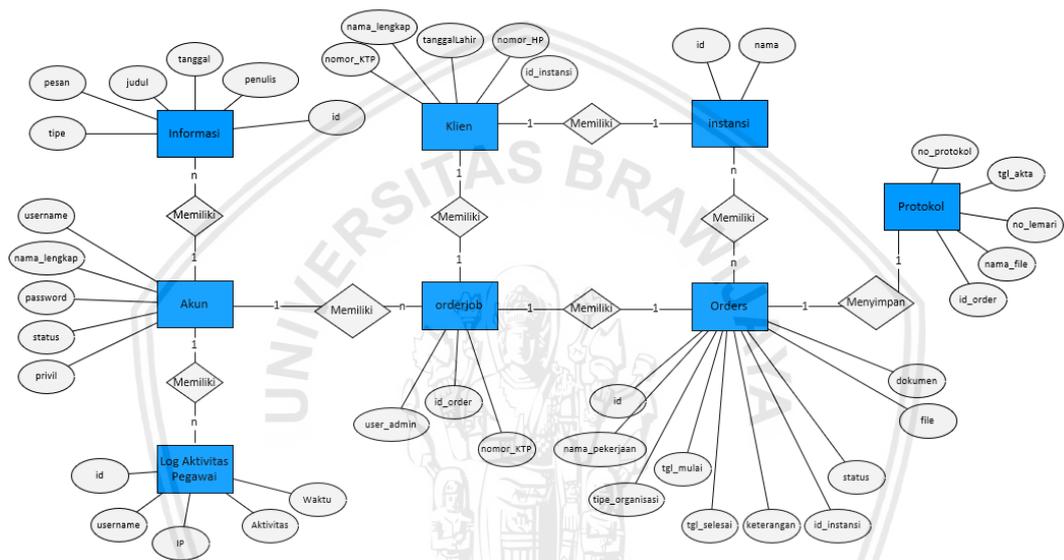
Tabel 5.3 Algoritme fungsi paper

1	mulai
2	mengambil data dari model akun di method getpaper tanpa parameter dan disimpan di variabel data
3	deklarasi variabel result
4	jika variabel data nilainya null
5	mencetak notifikasi tidak ada info kerusakan
6	else
7	melakukan looping foreach memecah variabel data ke dalam variabel key
8	jika variabel key dengan index tanggal selesai lebih dari tanggal saat ini
9	deklarasi variabel datetime1 dengan nilai tanggal saat ini
10	deklarasi datetime2 dengan nilai variabel key index tanggal selesai
11	deklarasi variabel deifference dengan nilai perbedaan antara datetime 2 dengan datetimel
12	jika variabel difference index y lebih dari sama dengan 5 atau index y sama dengan 3 atau index y sama dengan 1
13	deklarasi variabel nama dengan nilai variabel key index tipe organisasi ditambah variabel key index nama pekerjaan
14	deklarasi variabel lemari dengan nilai variabel key nomer lemari
15	menambahkan isi variabel result
16	jika variabel result nilai sama dengan kosong
17	mencetak tidak ada <i>deadline</i>
18	else
19	mencetak variabel result
20	selesai
21	

5.1.4 Perancangan Basis Data

Perancangan basis data pada sistem pembukuan dan administrasi protokol notaris ini digambarkan dalam Gambar 5.9 yang menggambarkan *Entity Relationship Diagram* (ERD) fase perancangan dari Sistem Pembukuan dan Administrasi Protokol Notaris. Dimana atas ERD ini diperoleh delapan entitas, yakni informasi, akun, log_aktivitas_pegawai, klien, orderjob, instansi, orders, dan

protokol. Entitas informasi mempunyai 6 atribut, entitas ini berkaitan terhadap entitas akun. Entitas akun mempunyai 5 atribut, entitas ini berkaitan dengan informasi, log_aktivitas_pegawai dan orderjob. Entitas orderjob mempunyai 3 atribut, entitas ini mempunyai kaitan dengan entitas klien, akun, dan orders. Entitas log_aktivitas_pegawai mempunyai 5 atribut, entitas ini berkaitan terhadap entitas akun. Entitas klien mempunyai 5 atribut, dimana entitas ini berkaitan terhadap entitas orderjob dan instansi. Entitas orders mempunyai 11 atribut dimana entitas ini berkaitan dengan orderjob, instansi, dan protokol. Entitas instansi mempunyai 2 atribut yang mana entitas ini berkaitan dengan entitas orders dan entitas klien. Dan entitas protokol mempunyai 5 atribut dimana entitas ini berkaitan terhadap entitas orders.



Gambar 5.8 Entity Relationship Diagram (ERD)

Tabel 5.4 Struktur tabel informasi

Nama Atribut	Tipe	Lebar	Keterangan
Id	Int	11	Id sebagai <i>primary key</i> .
Penulis	Varchar	100	Penulis sebagai <i>foreign key</i> .
Tanggal	Date	-	Keterangan waktu tanggal user.
Judul	Varchar	200	Menyimpan judul user.
Pesan	Teks	-	Menyimpan pesan user.
Tipe	Set	('Update', 'Informasi', 'Peringatan')	Menyimpan tipe masukan informasi user.

Pada tabel 5.4 menerangkan struktur tabel informasi yang teurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user teurai atas beberapa atribut antara lain, yakni id, penulis, tanggal, judul, pesan, dan tipe.

Tabel 5.5 Struktur tabel akun

Nama Atribut	Tipe	Lebar	Keterangan
Username	Varchar	50	Username sebagai <i>primary key</i> .
Nama_lengkap	Varchar	100	Menyimpan nama lengkap user.
Password	Varchar	200	Menyimpan <i>password</i> user.
Status	Tinyint	1	Keterangan status user.
Privil	Tinyint	1	Keterangan privil wewenang dari user.

Pada tabel 5.5 menerangkan struktur tabel akun yang teurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user teurai atas beberapa atribut antara lain, yakni username, nama_lengkap, password, status, dan privil.

Tabel 5.6 Struktur tabel log_aktivitas_pegawai

Nama Atribut	Tipe	Lebar	Keterangan
id	Int	11	Id sebagai <i>primary key</i> .
username	Varchar	50	Username sebagai <i>foreign key</i> .
IP	Varchar	25	Menyimpan IP user.
Aktivitas	Varchar	100	Menyimpan aktivitas user.
Waktu	Timestamp		Keterangan waktu akses user.

Pada tabel 5.6 menerangkan struktur tabel log_aktivitas_pegawai yang teurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user teurai atas beberapa atribut antara lain, yakni id, username, IP, aktivitas, dan waktu.

Tabel 5.7 Struktur tabel klien

Nama Atribut	Tipe	Lebar	Keterangan
Nomor_KTP	Varchar	16	Nomor_KTP sebagai <i>primary key</i> .

Tabel 5.7 Struktur tabel klien (lanjutan)

Nama Atribut	Tipe	Lebar	Keterangan
Nama_lengkap	Varchar	100	Menyimpan nama lengkap user.
TanggalLahir	Date	-	Menyimpan tanggal lahir user.
Nomor_HP	Varchar	14	Menyimpan nomor HP user.
Id_instansi	Int	11	Id_instansi sebagai <i>foreign key</i> .

Pada tabel 5.7 menerangkan struktur tabel klien yang terurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user terurai atas beberapa atribut antara lain, yakni nomor_KTP, nama_lengkap, tanggalLahir, nomor_HP, id_instansi.

Tabel 5.8 Struktur tabel orderjob

Nama Atribut	Tipe	Lebar	Keterangan
User_admin	Int	10	Id_user sebagai <i>primary key</i> .
Id_order	Varchar	100	Id_order sebagai <i>foreign key</i> .
Nomor_KTP	Varchar	100	Id_order sebagai <i>foreign key</i> .

Pada tabel 5.8 menerangkan struktur tabel orderjob yang terurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user terurai atas beberapa atribut antara lain, yakni user_admin, id_order, dan nomor_KTP.

Tabel 5.9 Struktur tabel instansi

Nama Atribut	Tipe	Lebar	Keterangan
Id	Int	11	Id_user sebagai <i>primary key</i> .
Nama	Varchar	100	Untuk menyimpan nama user.

Pada tabel 5.9 menerangkan struktur tabel instansi yang terurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user terurai atas beberapa atribut antara lain, yakni id dan nama.

Tabel 5.10 Struktur tabel orders

Nama Atribut	Tipe	Lebar	Keterangan
Id	Int	11	Id_user sebagai <i>primary key</i> .

Tabel 5.10 Struktur tabel orders (lanjutan)

Nama Atribut	Tipe	Lebar	Keterangan
Nama_pekerjaan	Varchar	100	Menyimpan nama pekerjaan user.
Tipe_organisasi	Set	('PT', 'Perkumpulan', 'Yayasan', 'Wasiat', 'Fidusia')	Menyimpan tipe masukan organisasi user.
Tgl_mulai	Date	-	Menyimpan tanggal mulai user.
Tgl_selesai	Date	-	Menyimpan tanggal selesai user.
Keterangan	Varchar	100	Menyimpan keterangan user.
Id_instansi	Int	11	Id_instansi sebagai <i>foreign key</i> .
Status	Int	11	Keterangan status user
File	Varchar	200	Menyimpan <i>file</i> user.
Dokumen	Dokumen	200	Menyimpan dokumen user.

Pada tabel 5.10 menerangkan struktur tabel user yang terdiri atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user terurai atas beberapa atribut antara lain, yakni id, nama_pekerjaan, tipe_organisasi, tgl_mulai, tgl_selesai, keterangan, id_instansi, status, file, dan dokumen.

Tabel 5.11 Struktur tabel protokol

Nama Atribut	Tipe	Lebar	Keterangan
No_protokol	Int	10	No_protokol sebagai <i>primary key</i> .
Tgl_akta	Date	-	Menyimpan tanggal akta user.
No_lemari	Varchar	10	Menyimpan nomor lemari user.
Id_order	Int	11	Id_order sebagai <i>foreign key</i> .
Nama_file	Varchar	200	Menyimpan nama file user.

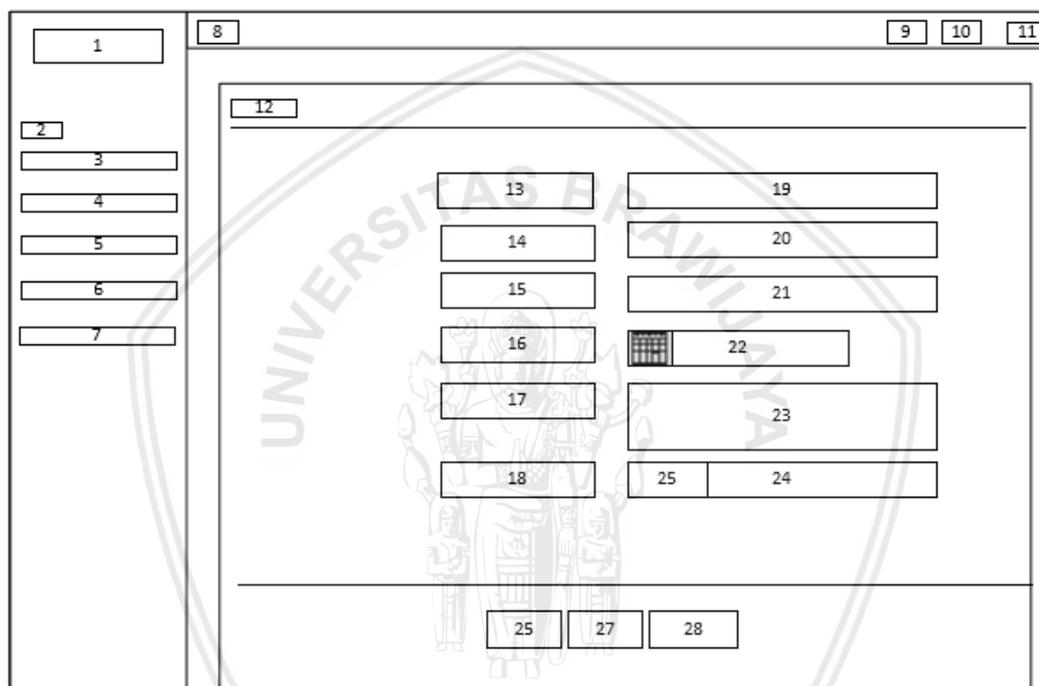
Pada tabel 5.11 menerangkan struktur tabel user yang terurai atas nama atribut, tipe, lebar dan keterangan dari setiap atribut. Tabel user terurai atas

beberapa atribut antara lain, yakni no_protokol, tgl_akta, no_lemari, id_order, dan nama_file.

5.1.5 Perancangan Antarmuka

Hasil perancangan antarmuka halaman antarmuka dari sistem pembukuan dan administrasi protokol notaris yang dibangun akan diterangkan pada perancangan antarmuka. Sampel antarmuka yang akan dicontohkan akan dicantumkan enam buah sampel. Dalam Gambar 5.10 dan Gambar 5.15 akan diutarakan hasil dari perancangan antarmuka sistem ini.

5.1.5.1 Perancangan Antarmuka Halaman Menambah Order Organisasi



Gambar 5.9 Perancangan antarmuka halaman menambah order organisasi

Tabel 5.12 Penjelasan perancangan antarmuka halaman menambah order organisasi

No.	Nama Objek	Tipe	Keterangan
1.	Logo kantor notaris dan teks notaris dewi	Logo text	Menunjukkan logo kantor notaris dan teks notaris dewi.
2.	Label menu	Label	Menunjukkan label menu.
3.	Simbol navigasi, dropdown, dan teks organisasi	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu organisasi.
4.	Simbol navigasi, dropdown, teks wasiat	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu wasiat.

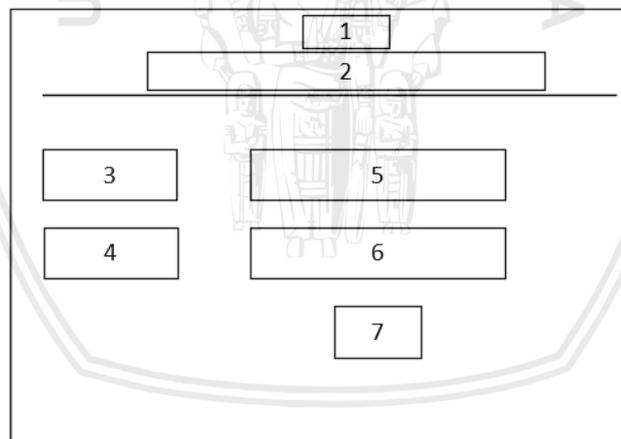
Tabel 5.12 Penjelasan perancangan antarmuka halaman menambah order (lanjutan)

No.	Nama Objek	Tipe	Keterangan
5.	Simbol navigasi, <i>dropdown</i> , dan teks fidusia	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu fidusia.
6.	Simbol navigasi dan teks klien	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman dari menu klien.
7.	Simbol navigasi dan teks protokol notaris	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman protokol notaris.
8.	Simbol navigasi menutup menu.	<i>Navigation symbol</i>	Untuk menutup menu.
9.	Simbol navigasi notifikasi <i>deadline</i> .	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>deadline</i> .
10.	Simbol navigasi notifikasi umur kertas.	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>umur kertas</i> .
11.	Simbol navigasi dan teks nama user.	<i>Navigation symbol dan text</i>	Untuk menampilkan sub-menu <i>logout</i> .
12.	Label tambah order.	Label	Untuk menunjukan label tambah order.
13.	Label tipe organisasi.	Label	Untuk menunjukan label tipe organisasi.
14.	Label nama organisasi.	Label	Untuk menunjukan label nama organisasi.
15.	Label tanggal mulai – tanggal selesai.	Label	Untuk menunjukan label tanggal mulai-tanggal selesai.
16.	Label keterangan tambahan.	Label	Untuk menunjukan label keterangan tambahan.
17.	Label dokumen.	Label	Untuk menunjukan label dokumen.
18.	Kolom klien	<i>Dropdown dan text</i>	Untuk menunjukan data klien.
19.	Teks <i>no file choosen</i>	Teks	Untuk menunjukan teks <i>no file choosen</i>
20.	Kolom tipe organisasi	<i>Dropdown dan text</i>	Untuk menunjukan data tipe organisasi.
21.	Kolom tanggal mulai – tanggal selesai.	<i>Datepicker</i>	Untuk menambahkan tanggal mulai dan tanggal selesai.

Tabel 5.12 Penjelasan perancangan antarmuka halaman menambah order (lanjutan)

No.	Nama Objek	Tipe	Keterangan
22.	Kolom keterangan tambahan.	Text area	Untuk menambahkan keterangan tambahan.
23.	Teks <i>no file choosen</i>	Text	Untuk menunjukan teks <i>no file choosen</i>
24.	Tombol <i>choose file</i>	Button	Untuk menambahkan <i>file</i> .
25.	Kolom dokumen.	Label	Untuk menunjukan teks <i>no file choosen</i> .
26.	Tombol batal	Button	Untuk membatalkan tambah order.
27.	Tombol <i>reset</i>	Button	Untuk mengatur ulang tambah order.
28.	Tombol tambah	Button	Untuk menambahkan data order.

5.1.5.2 Perancangan Antarmuka Halaman Login



Gambar 5.10 Perancangan antarmuka halaman login

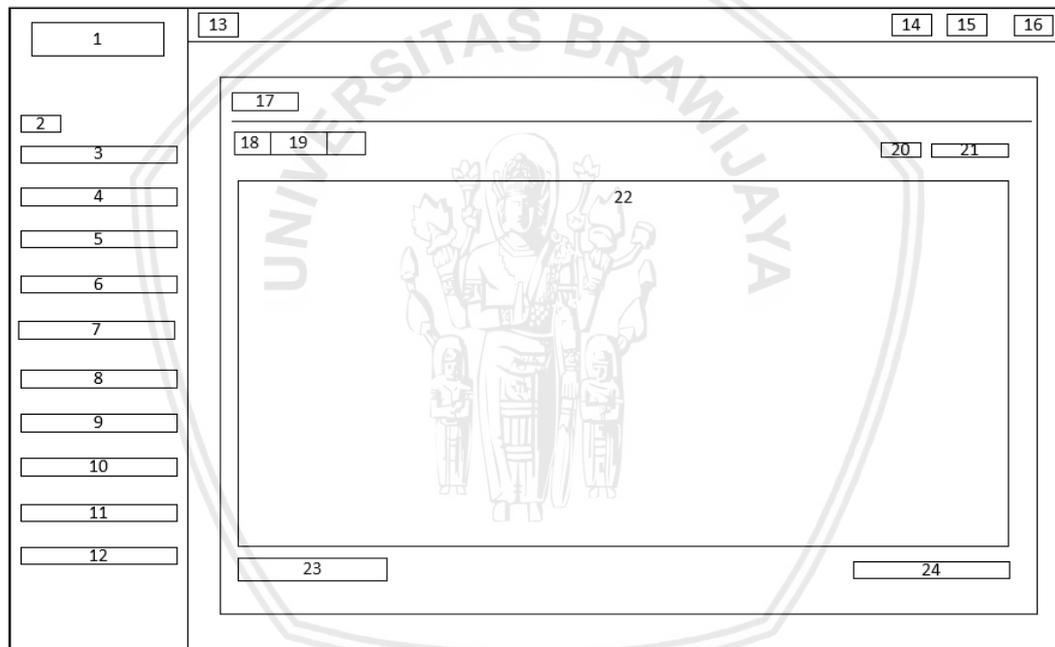
Tabel 5.13 Penjelasan perancangan antarmuka halaman login

No.	Nama Objek	Tipe	Keterangan
1.	Label <i>login</i>	Label	Untuk menunjukan label <i>login</i> .
2.	Label Kantor Notaris Dewi Andriani, S.H., M.H.	Label	Untuk menunjukan label Kantor Notaris Dewi Andriani, S.H., M.H.
3.	Label <i>username</i>	Label	Untuk menunjukan label <i>username</i> .

Tabel 5.13 Penjelasan perancangan antarmuka halaman login (lanjutan)

No.	Nama Objek	Tipe	Keterangan
4.	Label <i>password</i>	Label	Untuk menunjukkan label <i>password</i> .
5.	Kolom <i>username</i>	Teks	Kolom untuk masukan <i>username</i>
6.	Kolom <i>password</i>	Teks	Kolom untuk masukan <i>password</i> .
7.	Tombol <i>login</i>	<i>Button</i>	Tombol untuk melakukan <i>login</i> .

5.1.5.3 Perancangan Antarmuka Halaman Melihat Monitoring Kerja



Gambar 5.11 Perancangan antarmuka halaman melihat monitoring kerja

Tabel 5.14 Penjelasan perancangan antarmuka halaman melihat monitoring kerja

No.	Nama Objek	Tipe	Keterangan
1.	Logo kantor notaris dan teks notaris dewi	Logo text	Menunjukkan logo kantor notaris dan teks notaris dewi.
2.	Label menu	Label	Menunjukkan label menu.
3.	Simbol navigasi dan teks <i>dashboard</i>	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman <i>dashboard</i> .

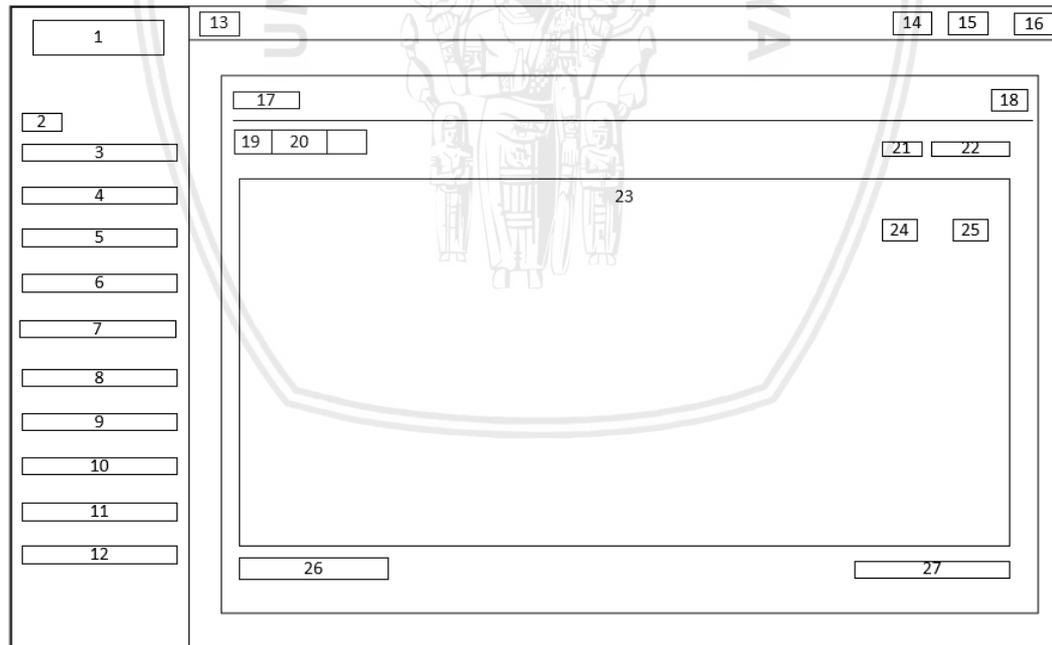
Tabel 5.14 Penjelasan perancangan antarmuka halaman melihat monitoring kerja (lanjutan)

No.	Nama Objek	Tipe	Keterangan
4.	Simbol navigasi, <i>dropdown</i> , dan teks organisasi	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu organisasi.
5.	Simbol navigasi, <i>dropdown</i> , teks wasiat	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu wasiat.
6.	Simbol navigasi, <i>dropdown</i> , dan teks fidusia	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu fidusia.
7.	Simbol navigasi dan teks klien	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman dari menu klien.
8.	Simbol navigasi dan teks protokol notaris	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman protokol notaris.
9.	Simbol navigasi dan teks akun pegawai	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman akun pegawai.
10.	Simbol navigasi dan teks informasi	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman informasi.
11.	Simbol navigasi dan teks monitoring kerja	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman monitoring kerja.
12.	Simbol navigasi dan teks log aktivitas pegawai	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman log aktivitas pegawai.
13.	Simbol navigasi menutup menu.	<i>Navigation symbol</i>	Untuk menutup menu.
14.	Simbol navigasi notifikasi <i>deadline</i> .	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>deadline</i> .
15.	Simbol navigasi notifikasi umur kertas.	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>umur kertas</i> .
16.	Simbol navigasi dan teks nama user.	<i>Navigation symbol dan text</i>	Untuk menampilkan sub-menu <i>logout</i> .
17.	Label monitoring kerja	Label	Untuk menunjukan label monitoring kerja.
18.	Teks <i>showing entries</i>	<i>Text</i>	Untuk menunjukan teks <i>showing entries</i> .
19.	Kolom <i>showing entries</i>	<i>Dropdown dan text</i>	Untuk menampilkan banyak data yang akan ditampilkan.

Tabel 5.14 Penjelasan perancangan antarmuka halaman melihat monitoring kerja (lanjutan)

No.	Nama Objek	Tipe	Keterangan
20.	Teks <i>search</i> :	<i>Text</i>	Untuk menunjukkan teks <i>search</i> .
21.	Kolom <i>search</i> :	<i>Dropdown dan text</i>	Untuk mencari data pada tabel monitoring kerja.
22.	Tabel data monitoring kerja.	<i>Table</i>	Untuk menampilkan data monitoring kerja.
23.	Teks <i>showing 1 to 10 of 50 entries</i>	<i>Text</i>	Untuk menunjukkan teks <i>showing 1 to 10 of 50 entries</i> .
24.	Tombol <i>previous 1 2 3 4 5 Next</i>	<i>Button</i>	Untuk menampilkan data sebelum, berikut, dan nomor tertentu data monitoring kerja.

5.1.5.4 Perancangan Antarmuka Halaman Melihat Klien



Gambar 5.12 Perancangan antarmuka halaman melihat klien

Tabel 5.15 Penjelasan perancangan antarmuka halaman melihat klien

No.	Nama Objek	Tipe	Keterangan
1.	Logo kantor notaris dan teks notaris dewi	<i>Logo text</i>	Menunjukkan logo kantor notaris dan teks notaris dewi.



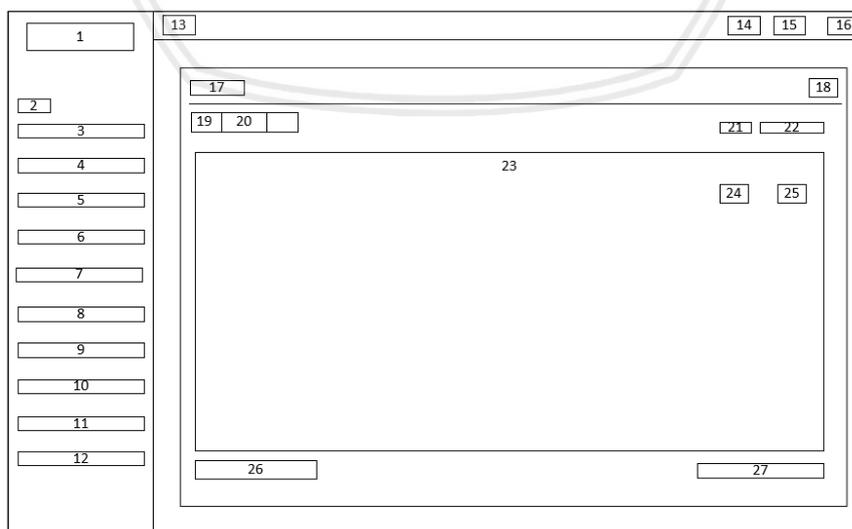
**Tabel 5.15 Penjelasan perancangan antarmuka halaman melihat klien
(lanjutan)**

No.	Nama Objek	Tipe	Keterangan
2.	Label menu	Label	Menunjukkan label menu.
3.	Simbol navigasi dan teks <i>dashboard</i>	<i>Navigation symbol and text</i>	Untuk menampilkan halaman <i>dashboard</i> .
4.	Simbol navigasi, <i>dropdown</i> , dan teks organisasi	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu organisasi.
5.	Simbol navigasi, <i>dropdown</i> , teks wasiat	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu wasiat.
6.	Simbol navigasi, <i>dropdown</i> , dan teks fidusia	<i>Navigation symbol, dropdown, dan text</i>	Untuk menampilkan sub-menu dari menu fidusia.
7.	Simbol navigasi dan teks klien	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman dari menu klien.
8.	Simbol navigasi dan teks protokol notaris	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman protokol notaris.
9.	Simbol navigasi dan teks akun pegawai	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman akun pegawai.
10.	Simbol navigasi dan teks informasi	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman informasi.
11.	Simbol navigasi dan teks monitoring kerja	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman monitoring kerja.
12.	Simbol navigasi dan teks log aktivitas pegawai	<i>Navigation symbol dan text</i>	Untuk menampilkan halaman log aktivitas pegawai.
13.	Simbol navigasi menutup menu.	<i>Navigation symbol</i>	Untuk menutup menu.
14.	Simbol navigasi notifikasi <i>deadline</i> .	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>deadline</i> .
15.	Simbol navigasi notifikasi umur kertas.	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>umur kertas</i> .
16.	Simbol navigasi dan teks nama user.	<i>Navigation symbol dan text</i>	Untuk menampilkan sub-menu <i>logout</i> .
17.	Label data klien	Label	Untuk menunjukkan label data klien.
18.	Tombol tambah	Button	Untuk menampilkan halaman tambah klien baru.

Tabel 5.15 Penjelasan perancangan antarmuka halaman melihat klien (lanjutan)

No.	Nama Objek	Tipe	Keterangan
19.	Teks <i>showing entries</i>	<i>Text</i>	Untuk menunjukkan teks <i>showing entries</i> .
20.	Kolom <i>showing entries</i>	<i>Dropdown dan text</i>	Untuk menampilkan banyak data yang akan ditampilkan.
21.	Teks <i>search:</i>	<i>Text</i>	Untuk menunjukkan teks <i>search</i> .
22.	Kolom <i>search:</i>	<i>Dropdown dan text</i>	Untuk mencari data pada tabel monitoring kerja.
23.	Tabel data klien.	<i>Table</i>	Untuk menampilkan data klien.
24.	Tombol lihat riwayat klien	<i>Button</i>	Untuk menampilkan data riwayat klien.
25.	Tombol ubah klien	<i>Button</i>	Untuk menampilkan halaman mengubah klien.
26.	Teks <i>showing 1 to 10 of 50 entries</i>	<i>Teks</i>	Untuk menunjukan teks <i>showing 1 to 10 of 50 entries</i> .
27.	Tombol <i>previous 1 2 3 4 5 Next</i>	<i>Button</i>	Untuk menampilkan data sebelum, berikut, dan nomor tertentu data monitoring kerja.

5.1.5.5 Perancangan Antarmuka Halaman Melihat Akun



Gambar 5.13 Perancangan antarmuka halaman melihat akun



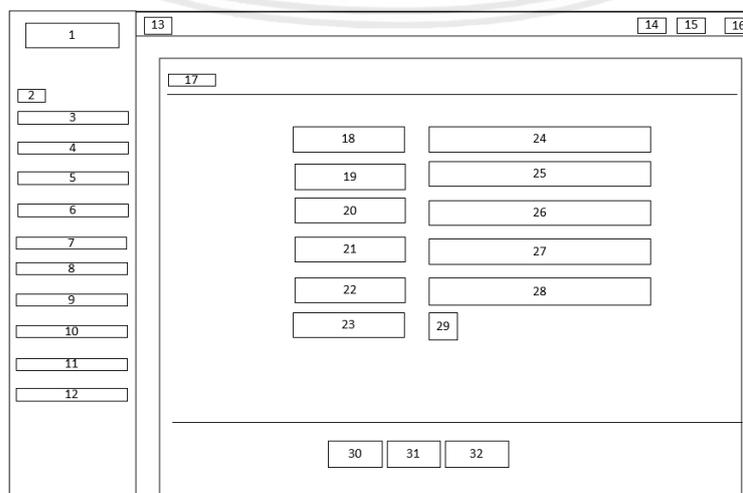
Tabel 5.16 Penjelasan perancangan antarmuka halaman melihat akun

No.	Nama Objek	Tipe	Keterangan
1.	Logo kantor notaris dan teks notaris dewi	Logo <i>text</i>	Menunjukkan logo kantor notaris dan teks notaris dewi.
2.	Label menu	Label	Menunjukkan label menu.
3.	Simbol navigasi dan teks <i>dashboard</i>	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman <i>dashboard</i> .
4.	Simbol navigasi, <i>dropdown</i> , dan teks organisasi	<i>Navigation symbol</i> , <i>dropdown</i> , dan <i>text</i>	Untuk menampilkan sub-menu dari menu organisasi.
5.	Simbol navigasi, <i>dropdown</i> , teks wasiat	<i>Navigation symbol</i> , <i>dropdown</i> , dan <i>text</i>	Untuk menampilkan sub-menu dari menu wasiat.
6.	Simbol navigasi, <i>dropdown</i> , dan teks fidusia	<i>Navigation symbol</i> , <i>dropdown</i> , dan <i>text</i>	Untuk menampilkan sub-menu dari menu fidusia.
7.	Simbol navigasi dan teks klien	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman dari menu klien.
8.	Simbol navigasi dan teks protokol notaris	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman protokol notaris.
9.	Simbol navigasi dan teks akun pegawai	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman akun pegawai.
10.	Simbol navigasi dan teks informasi	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman informasi.
11.	Simbol navigasi dan teks monitoring kerja	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman monitoring kerja.
12.	Simbol navigasi dan teks log aktivitas pegawai	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman log aktivitas pegawai.
13.	Simbol navigasi menutup menu.	<i>Navigation symbol</i>	Untuk menutup menu.
14.	Simbol navigasi notifikasi <i>deadline</i> .	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>deadline</i> .
15.	Simbol navigasi notifikasi umur kertas.	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>umur kertas</i> .
16.	Simbol navigasi dan teks nama user.	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan sub-menu <i>logout</i> .
17.	Label data akun	Label	Untuk menunjukan label data akun.

Tabel 5.16 Penjelasan perancangan antarmuka halaman melihat akun (lanjutan)

No.	Nama Objek	Tipe	Keterangan
18.	Tombol tambah	Button	Untuk menampilkan halaman tambah akun baru.
19.	Teks <i>showing entries</i>	Text	Untuk menunjukan teks <i>showing entries</i> .
20.	Kolom <i>showing entries</i>	Dropdown dan text	Untuk menampilkan banyak data yang akan ditampilkan.
21.	Teks <i>search:</i>	Teks	Untuk menunjukan teks <i>search</i> .
22.	Kolom <i>search:</i>	Dropdown dan text	Untuk mencari data pada tabel monitoring kerja.
23.	Tabel data akun.	Table	Untuk menampilkan data akun.
24.	Tombol ubah akun	Button	Untuk menampilkan halaman mengubah akun.
25.	Tombol hapus akun	Button	Untuk menghapus akun.
26.	Teks <i>showing 1 to 10 of 50 entries</i>	Text	Untuk menunjukan teks <i>showing 1 to 10 of 50 entries</i> .
27.	Tombol <i>previous 1 2 3 4 5 Next</i>	Button	Untuk menampilkan data sebelum, berikut, dan nomor tertentu data monitoring kerja.

5.1.5.6 Perancangan Antarmuka Halaman Membuat Akun



Gambar 5.14 Perancangan antarmuka halaman membuat akun



Tabel 5.17 Penjelasan perancangan antarmuka halaman membuat akun

No.	Nama Objek	Tipe	Keterangan
1.	Logo kantor notaris dan teks notaris dewi	<i>Logo text</i>	Menunjukkan logo kantor notaris dan teks notaris dewi.
2.	Label menu	Label	Menunjukkan label menu.
3.	Simbol navigasi dan teks <i>dashboard</i>	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman <i>dashboard</i> .
4.	Simbol navigasi, <i>dropdown</i> , dan teks organisasi	<i>Navigation symbol</i> , <i>dropdown</i> , dan <i>text</i>	Untuk menampilkan sub-menu dari menu organisasi.
5.	Simbol navigasi, <i>dropdown</i> , teks wasiat	<i>Navigation symbol</i> , <i>dropdown</i> , dan <i>text</i>	Untuk menampilkan sub-menu dari menu wasiat.
6.	Simbol navigasi, <i>dropdown</i> , dan teks fidusia	<i>Navigation symbol</i> , <i>dropdown</i> , dan <i>text</i>	Untuk menampilkan sub-menu dari menu fidusia.
7.	Simbol navigasi dan teks klien	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman dari menu klien.
8.	Simbol navigasi dan teks protokol notaris	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman protokol notaris.
9.	Simbol navigasi dan teks akun pegawai	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman akun pegawai.
10.	Simbol navigasi dan teks informasi	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman informasi.
11.	Simbol navigasi dan teks monitoring kerja	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman monitoring kerja.
12.	Simbol navigasi dan teks log aktivitas pegawai	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan halaman log aktivitas pegawai.
13.	Simbol navigasi menutup menu	<i>Navigation symbol</i>	Untuk menutup menu.
14.	Simbol navigasi notifikasi <i>deadline</i>	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>deadline</i> .
15.	Simbol navigasi notifikasi umur kertas	<i>Navigation symbol</i>	Untuk menampilkan notifikasi <i>umur kertas</i> .
16.	Simbol navigasi dan teks nama user	<i>Navigation symbol</i> dan <i>text</i>	Untuk menampilkan sub-menu <i>logout</i> .
17.	Label tambah akun	Label	Untuk menunjukkan label tambah akun.
18.	Label username	Label	Untuk menunjukkan label username.

Tabel 5.17 Penjelasan perancangan antarmuka halaman membuat akun (lanjutan)

No.	Nama Objek	Tipe	Keterangan
19.	Label nama lengkap	Label	Untuk menunjukkan label nama lengkap.
20.	Label <i>password</i>	Label	Untuk menunjukkan label <i>password</i> .
21.	Label re-type <i>password</i>	Label	Untuk menunjukkan label re-type <i>password</i> .
22.	Label hak akses	Label	Untuk menunjukkan label hak akses.
23.	Label status aktif.	Label	Untuk menunjukkan label status aktif.
24.	Kolom <i>username</i>	Text	Untuk menambahkan <i>username</i> .
25.	Kolom nama lengkap	Text	Untuk menambahkan nama lengkap.
26.	Kolom <i>password</i>	Text	Untuk menambahkan <i>password</i> .
27.	Kolom re-type <i>password</i>	Text	Untuk menambahkan re-type <i>password</i> .
28.	Kolom hak akses	<i>Dropdown</i> dan Text	Untuk menunjukkan data hak akses.
29.	Kolom status aktif	<i>Checkbox</i>	Untuk menambahkan status aktif.
30.	Tombol batal	Button	Untuk membatalkan tambah akun.
31.	Tombol <i>reset</i>	Button	Untuk mengatur ulang tambah akun.
32.	Tombol tambah	Button	Untuk menambahkan data akun.

5.2 Implementasi Sistem

5.2.1 Spesifikasi Pengembangan Sistem

Tercantum dua sub-bab untuk spesifikasi sistem pada sistem pembukuan dan administrasi protokol notaris yang digunakan, yakni spesifikasi perangkat keras dan spesifikasi perangkat lunak. Spesifikasi sistem pada sistem digunakan untuk mendukung prosedur pengembangan sistem.

5.2.1.1 Spesifikasi Perangkat Keras

Pada Tabel 5.18 merupakan spesifikasi perangkat keras yang diterapkan dalam membangun sistem pembukuan dan administrasi protokol notaris.

Tabel 5.18 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
<i>System Model</i>	ASUS ROG GL552JX
<i>Processor</i>	Intel® Core™ i7-4720HQ CPU @ 2.60GHz 2.59 GHz
<i>Main Memory</i>	RAM 12,00 GB (11,9 GB usable)
<i>Display</i>	NVIDIA GTX 950m
<i>Hard Disk</i>	1 TB
<i>Resolution</i>	1366x768

5.2.1.2 Spesifikasi Perangkat Lunak

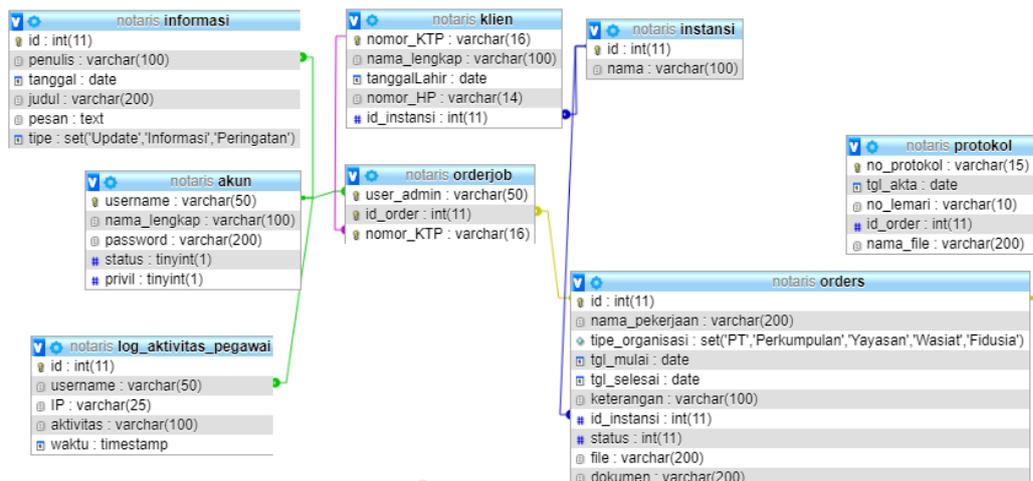
Pada Tabel 5.19 merupakan spesifikasi perangkat lunak yang diterapkan dalam membangun sistem pembukuan dan administrasi protokol notaris.

Tabel 5.19 Spesifikasi perangkat lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 Home 64-bit
IDE	Sublime Text 3
<i>Programming Editor</i>	Microsoft Visio 2016, draw.io
<i>Programing Language</i>	PHP, Javascript, CSS, HTML, AJAX
DBMS	Mysql
<i>Supporting System Application</i>	XAMPP v3.2.2 Microsoft Word 2016
<i>Browser</i>	Google Chrome

5.2.2 Implementasi Basis Data

Dasar dari perancangan basis data dalam Gambar 5.8 diimplementasikan melalui PHPMyAdmin 4.5.1 menjadi implementasi basis data. Dalam Gambar 5.16 menggambarkan skema basis data sistem pembukuan dan administrasi protokol notaris. Dalam skema basis data tersebut terdapat delapan tabel, yaitu informasi, akun, log_aktivitas_pegawai, klien, orderjob, instansi, orders, dan protokol.



Gambar 5.15 Skema basis data sistem pembukuan dan administrasi protokol notaris

5.2.3 Implementasi Kode Program

Produk perancangan algoritme yang sudah dibuat ke dalam kode komputer dengan kode pemrograman dikonversikan menjadi implementasi kode program. Terdapat tiga buah kode program yang menjadi contoh, yakni kode program tambahOrder, work, dan paper.

5.2.3.1 Implementasi Kode Program tambahOrder

Fungsi tambahOrder ialah fungsi ada pada kelas order yang dipanggil saat pegawai melakukan tambah order organisasi. Fungsi tambahorder ini ditunjukkan pada Tabel 5.20.

Nama Kelas: Order

Nama Fungsi: tambahorder()

Algoritme:

Tabel 5.20 Kode program fungsi tambahOrder

```

1      public function tambahorder()
2      {
3          if($this->user['log'] != 'Pin') {
4              redirect(base_url('order/pesan'), 'refresh');
5          }else{
6              $data['mode'] = $this->user['log'];
7              $data['klien'] = $this->klien->getKlien();
8              $data['instansi'] = $this->order->getInstansi();
9              $data['username'] = $this->user['nama'];
10             $this->form_validation->set_rules('klien', 'Nama
klien', 'required');
11
12             if ($this->form_validation->run() === FALSE)
13             {
14                 $this->load->view('header');
15                 $this->load->view('nav', $data);
16                 $this->load->view('v_buatOrder');

```



Tabel 5.20 Kode program fungsi tambahOrder (lanjutan)

```

17 $this->load->view('footer');
18     }
19     else
20     {
21         $config['upload_path'] =
22         './dokumen/order';
23         $config['allowed_types'] = 'pdf';
24         $config['max_size'] = 1024 * 12;
25         $this->load->library('upload', $config);
26         if (!$this->upload->do_upload('dokumen'))
27         {
28             echo $this->upload->display_errors();
29         } else {
30             $data = $this->upload->data();
31             $this->order-
32             >daftar($data['file_name']);
33         }
34         redirect(base_url('order/pesan'),'refresh');
35     }
36 }
37 }
39 }

```

5.2.3.2 Implementasi Kode Program Work

Fungsi work merupakan algoritme yang menjelaskan proses ketika pemilik dan pegawai mendapatkan notifikasi tenggat dari pekerjaan yang sedang dikerjakan. Fungsi work ini ditunjukkan pada Tabel 5.21.

Nama Kelas: Helper

Nama Fungsi: work()

Algoritme:

Tabel 5.21 Kode program fungsi work

```

1 public function work()
2     {
3         if ($this->user['log'] == 'Nin') {
4             $data = $this->akun->getwork();
5         } else {
6             $data = $this->akun->getwork($this->user['username']);
7         }
8         $res = '';
9         if($data == null) {
10            echo '<li><a><span><span>Tidak ada
11            deadline</span></span></a></li>';
12        } else {
13            foreach ($data as $key) {
14                if (strtotime($key['tgl_selesai']) >
15                strtotime(date("Y-m-d"))) {
16                    $datetime1 = new DateTime();
17                    $datetime2 = new DateTime($key['tgl_selesai']);
18                    $difference = $datetime1->diff($datetime2);
19                    if ($difference->days <= 7) {
20                        $nama = $key['nama_klien'];
21                        $res .= '<li><a>
22                        <span>

```

Tabel 5.21 Kode program fungsi work (lanjutan)

```

22         <span>Deadline pekerjaan dari
    '.$nama.'</span>
23         </span>
24     </span>
25     <span class="message">
26         <b>'. $difference->days.' hari lagi</b>
27     </span>
28     </a></li>';
29     }
30 }
31 }
32     if ($res == '') {
33         echo '<li><a><span><span>Tidak ada
deadline</span></span></a></li>';
34     } else {
35         echo $res;
36     }
37 }
38 }

```

5.2.3.3 Implementasi Kode Program Paper

Fungsi paper adalah algoritme yang menerangkan ketika pemilik dan pegawai mendapatkan notifikasi umur kertas dari protokol notaris yang telah disimpan di dalam *database*. Fungsi monitor ini tercantum pada Tabel 5.22.

Nama Kelas: Helper

Nama Fungsi: paper()

Algoritme:

Tabel 5.22 Kode program fungsi paper

```

1 public function paper()
2 {
3     $data = $this->akun->getpaper();
4     $res = '';
5     if($data == null) {
6         echo '<li><a><span><span>Tidak ada info
kerusakan</span></span></a></li>';
7     } else {
8         foreach ($data as $key) {
9             if (strtotime($key['tgl_akta']) < strtotime(date("Y-
m-d"))) {
10                $datetime1 = new DateTime();
11                $datetime2 = new DateTime($key['tgl_akta']);
12                $difference = $datetime1->diff($datetime2);
13                if ($difference->y >= 5 || $difference->y == 3
|| $difference->y == 1) {
14                    $nama = $key['tipe_organisasi'].
"$key['nama_pekerjaan']";
16                    $lemari = $key['no_lemari'];
17                    $res .= '<li><a>
18                        <span>
19                            <span>Kertas untuk berkas '.$nama.' harap
diganti pada lemari '.$lemari.'</span>
20                        </span>
21                        <span class="message">
22                            <b>'. $difference->y.' tahun lebih</b>
22                        </span>
23                    </a></li>';

```

Tabel 5.22 Kode program fungsi paper (lanjutan)

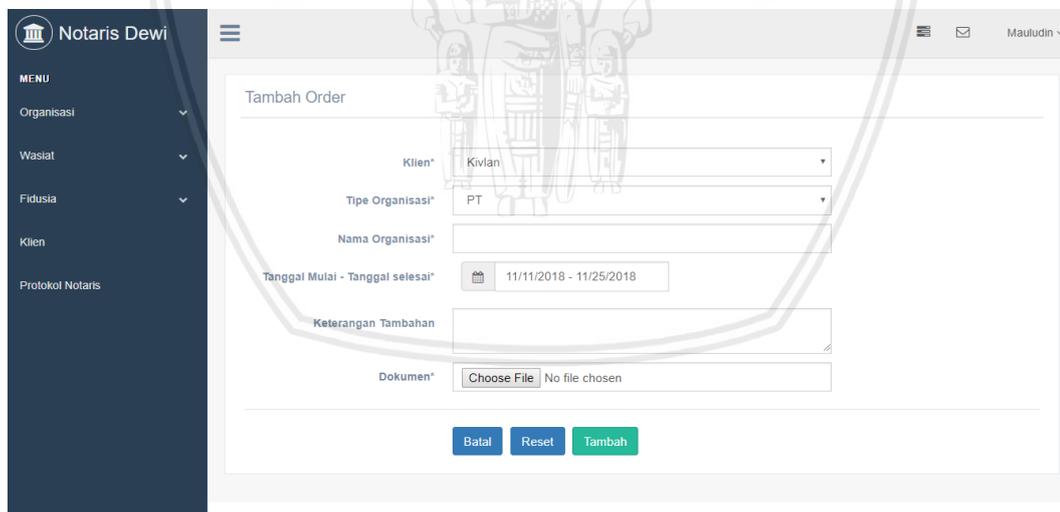
```

24         }
25     }
26 }
27     if ($res == '') {
28         echo '<li><a><span><span>Tidak ada info
29 kerusakan</span></span></a></li>';
30     } else {
31         echo $res;
32     }
33 }
34 }
35 }
    
```

5.2.4 Implementasi Antarmuka

5.2.4.1 Implementasi Antarmuka Menambah Pesan Nama Organisasi

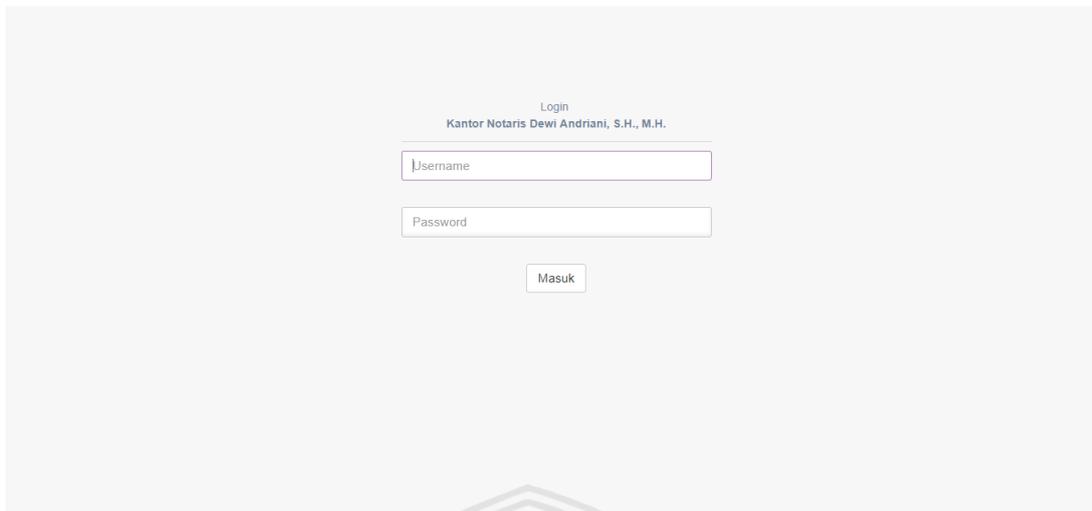
Pada implementasi antarmuka menambah pesan nama organisasi yang ditunjukkan dalam Gambar 5.16, terdapat form yang dapat diisi oleh pengguna ketika ingin membuat pesan nama organisasi. Pengguna harus mengisi klien, tipe organisasi, nama organisasi, tanggal mulai dan selesai, dan unggah dokumen. Kemudian keterangan tambahan sebagai opsi. Lalu pengguna dapat menekan tombol tambah untuk menambah pesan nama organisasi, reset untuk mengosongkan isian yang telah diisi oleh penggunan, dan batal jika ingin membatalkan.



Gambar 5.16 Implementasi antarmuka menambah pesan nama organisasi

5.2.4.2 Implementasi Antarmuka Login

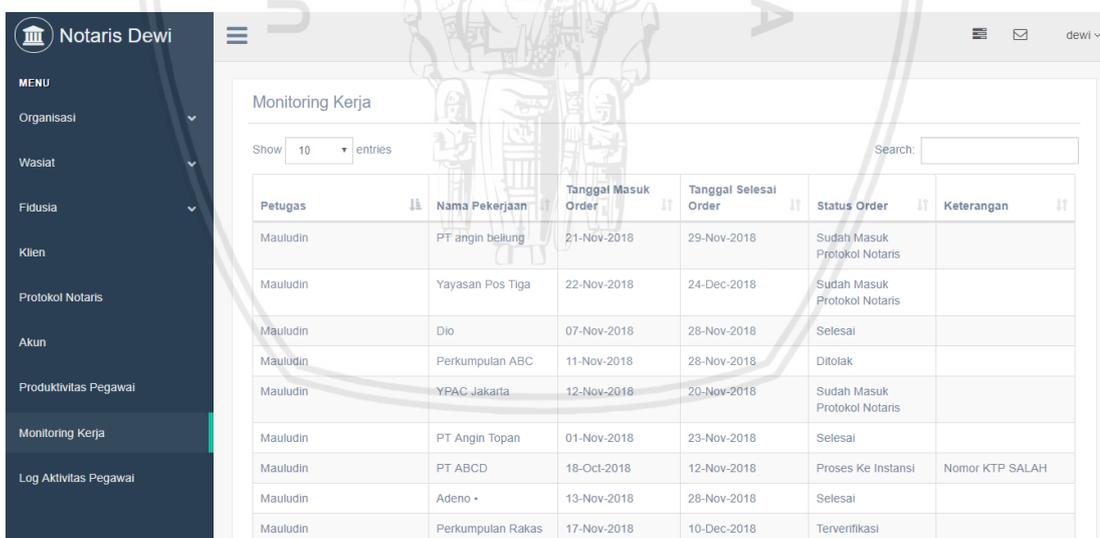
Dalam Gambar 5.17 ditunjukkan implementasi antarmuka login. Kolom *Username dan password* perlu diisi oleh pengguna pada *form* yang telah diperoleh. Kemudian terdapat tombol login yang digunakan untuk masuk ke dalam sistem sebagai pengguna yang telah terautentifikasi.



Gambar 5.17 Implementasi antarmuka login

5.2.4.3 Implementasi Antarmuka Melihat Monitoring Kerja

Dalam Gambar 5.18 menerangkan implementasi antarmuka melihat monitoring kerja. Terdapat tabel untuk menunjukkan informasi yang merupakan detail dari monitoring kerja. Informasi yang terkandung adalah petugas, nama pekerjaan, tanggal masuk order, tanggal selesai order, status order, dan keterangan.

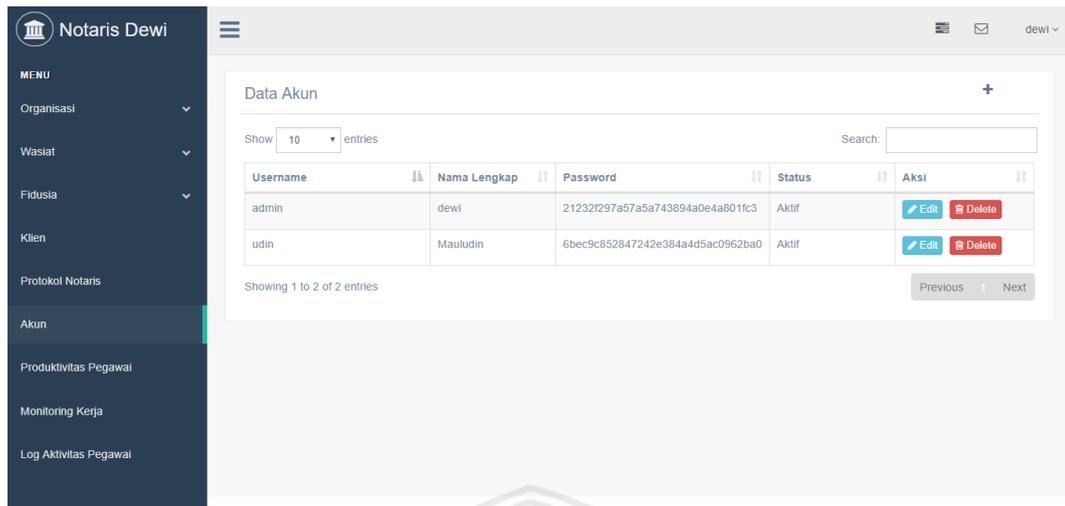


Gambar 5.18 Implementasi antarmuka melihat monitoring kerja

5.2.4.4 Implementasi Antarmuka Melihat Akun

Pada implementasi antarmuka melihat akun yang ditunjukkan dalam Gambar 5.20, terdapat tabel untuk menunjukkan informasi akun yang telah ada pada sistem ini. Informasi yang terkandung adalah *username*, nama lengkap, *password*, status aktif, selesai order, dan status order. Lalu pengguna dapat menekan tombol *edit* untuk mengubah akun yang telah ada dan hapus untuk menghapus akun yang telah ada.

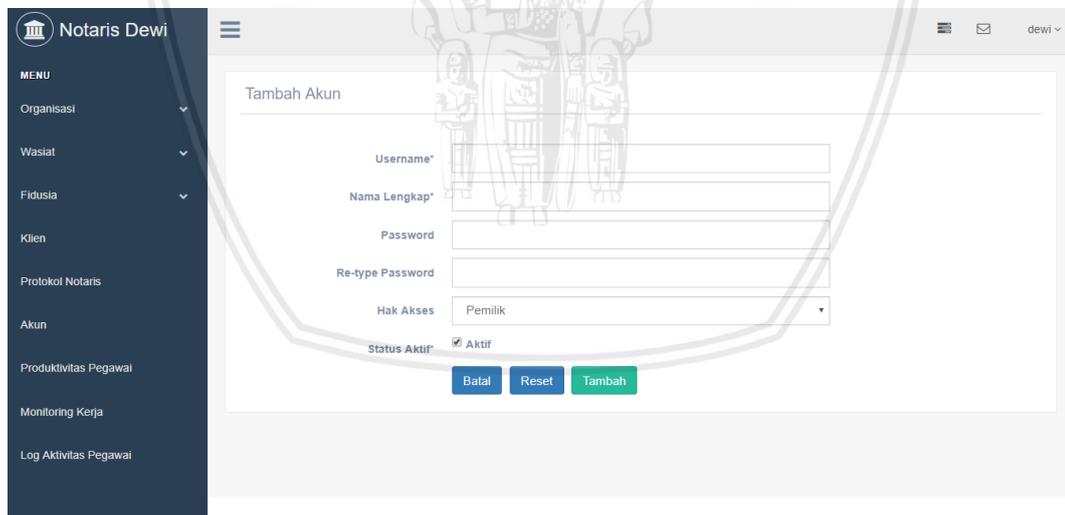




Gambar 5.19 Implementasi antarmuka melihat akun

5.2.4.5 Implementasi Antarmuka Membuat Akun

Pada implementasi antarmuka menambah akun yang ditunjukkan dalam Gambar 5.21, terdapat form yang dapat diisi oleh pengguna saat membuat akun. *Username*, nama lengkap, *password*, *re-type password*, hal akses, dan status aktif perlu diisi oleh pengguna. Lalu pengguna dapat menekan tombol tambah untuk menambah akun, reset untuk mengosongkan isian yang telah diisi oleh pengguna, dan batal jika ingin membatalkan tambah akun.



Gambar 5.20 Implementasi antarmuka membuat akun

BAB 6 PENGUJIAN

Dalam penelitian yang telah dilakukan, pengujian sistem akan dibahas pada bab ini. Bagian pengujian dilakukan selepas diselesaikannya implementasi dari sistem. Pengujian ini didasarkan pada kebutuhan fungsional dan non-fungsional sistem yang sudah didefinisikan. Hasil dari rekayasa kebutuhan pengujian ini akan dijadikan acuan dalam penarikan kesimpulan dan saran Sistem Pembukuan dan Administrasi Protokol Notaris. Pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian usabilitas akan membagi pembahasan bab ini.

6.1 Pengujian Unit

Pengujian unit menjelaskan mengenai rincian pengujian terhadap setiap komponennya. Pengujian tersebut bertujuan untuk menguji kebenaran komponen dari kode programnya. Metode yang digunakan pada pengujian ini adalah pengujian unit. Pada penelitian ini terdapat tiga fungsi yang akan dibahas. Fungsi tersebut antara lain fungsi tambahOrder, fungsi work, dan fungsi paper. Tabel 6.1, Tabel 6.3, dan Tabel 6.5 merupakan tiga algoritma fungsi yang menunjukkan hasil pemaparan perancangan algoritme.

6.1.1 Pengujian Unit Kelas Order pada Fungsi tambahOrder()

1. Pseudocode

Tabel 6.1 Pseudocode tambahOrder

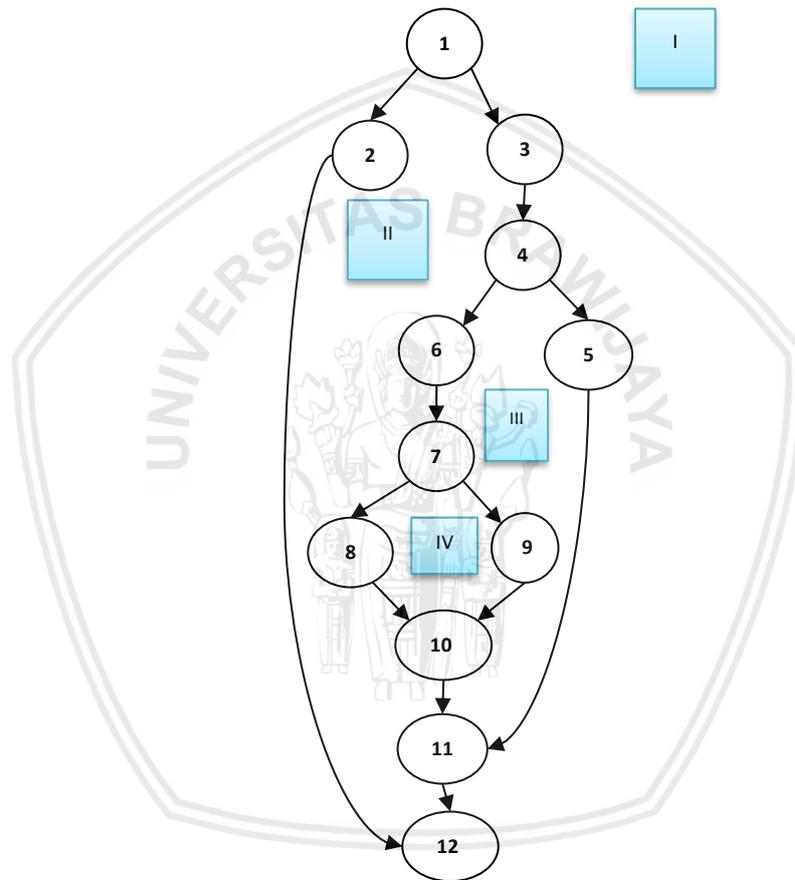
1	IF (user[log] != Pin) THEN	1
2	redirect(base_url(order/pesan), refresh)	2
3	ELSE THEN	
4	Inisialisasi variabel data[mode] = user[log]	3
5	Inisialisasi variabel data[klien] = getKlien()	
6	Inisialisasi variabel data[instansi] = getInstansi()	
7	Inisialisasi variabel data[username] = user[nama]	
8	form_validation -> set_rules(klien, nama klien, required)	
9	IF (form_validation === FALSE) THEN	4
10	Menampilkan halaman header	5
11	Menampilkan halaman nav	
12	Menampilkan halaman buat order	
13	Menampilkan halaman footer	
14	ELSE THEN	
15	Inisialisasi[upload_path] = ./dokumen/order	6
16	Inisialisasi[allowed_types] = pdf	
17	Inisialisasi[max_size] = 1024 * 12	
18	Memasukan library upload	
19	IF (!upload->do_upload(dokumen)) THEN	7
20	tampilkan pesan error	8
21	ELSE THEN	
22	Inisialisasi variabel data = data()	9
23	memasukan data ke dalam database	
24	redirect(base_url(order/pesan), refresh)	

Tabel 6.1 Pseudocode tambahOrder (lanjutan)

25	END IF ————— (10)
26	END IF ————— (11)
27	END IF ————— (12)

2. *Basic Path Testing*

a) *Flow Graph*



Gambar 6.1 Flow graph fungsi tambahOrder

b) *Cyclomatic Complexity V(G)*

- I. $V(G) = R = 4$
- II. $V(G) = E - N + 2 = 14 - 12 + 2 = 4$
- III. $V(G) = P + 1 = 3 + 1 = 4$

c) *Independent Path*

- I. Jalur 1 = 1-2-12
- II. Jalur 2 = 1-3-4-5-11-12
- III. Jalur 3 = 1-3-4-6-7-8-10-11-12

IV. Jalur 4 = 1-3-4-6-7-9-10-11-12

Tabel 6.2 Hasil pengujian unit fungsi tambahOrder() kelas Order

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1-2-12	Kelas <i>Driver</i> memanggil fungsi <i>tambahOrder()</i> dengan <i>log user</i> adalah bukan pegawai.	Kembali ke halaman pesan nama organisasi	Kembali ke halaman pesan nama organisasi.	Valid.
2	1-3-4-5-11-12	Kelas <i>Driver</i> memanggil fungsi <i>tambahOrder()</i> dengan <i>log user</i> adalah pegawai, menginisialisasi data ke dalam variabel, dan tidak mengisi form.	Menampilkan halaman membuat pesan nama organisasi.	Menampilkan halaman membuat pesan nama organisasi.	Valid.
3	1-3-4-6-7-8-10-11-12	Kelas <i>Driver</i> memanggil fungsi <i>tambahOrder()</i> dengan <i>log user</i> adalah pegawai, menginisialisasi data ke dalam variabel, mengisi form, dan tidak menjalankan fungsi <i>do_upload</i> .	Mencetak pesan <i>error</i> .	Mencetak pesan <i>error</i> .	Valid.
4	1-3-4-6-7-9-10-11-12	Kelas <i>Driver</i> memanggil fungsi <i>tambahOrder()</i> dengan <i>log user</i> adalah pegawai, menginisialisasi data ke dalam variabel, mengisi form, menjalankan fungsi <i>do_upload</i> , mengunggah data, dan memasukan masukan dengan fungsi <i>daftar(\$data['filename'])</i> .	Menyimpan masukan dengan fungsi daftar ke dalam <i>database</i> dan menampilkan halaman pesan nama organisasi.	Menyimpan masukan dengan fungsi daftar ke dalam <i>database</i> dan menampilkan halaman pesan nama organisasi.	Valid.

6.1.2 Pengujian Unit Kelas Helper pada Fungsi Work()

1. Pseudocode

Tabel 6.3 Pseudocode work

1	IF (user[log] == Nin) THEN	1
2	inisialisasi variabel data = getwork()	2
3	ELSE THEN	
4	inisialisasi variabel data = getwork(user[username])	3
5	END IF	4
6	inisialisasi variabel res = ''	5
7	IF(data == null) THEN	6
8	mencetak Tidak ada <i>deadline</i>	7
10	ELSE THEN	
11	FOREACH (data as key)	8

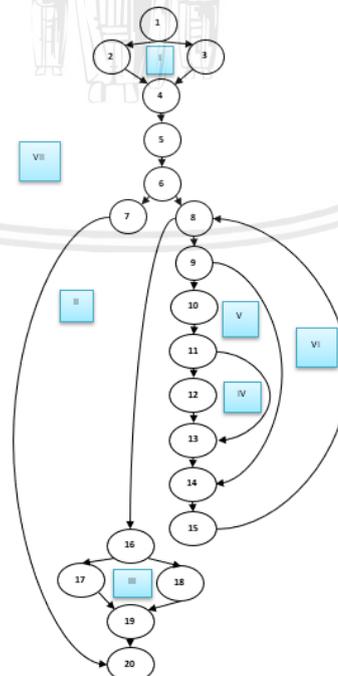


Tabel 6.3 Pseudocode work (lanjutan)

12	IF (strtotime(key[tgl_selesai]) > strtotime	9
	(date(Y-m-d))) THEN	
13	inisialisasi variabel datetime1 = new	10
	DateTime()	
14	inisialisasi variabel datetime2 = new	
	DateTime(key[tgl_selesai])	
15	inisialisasi variabel difference =	
	\$datetime1->diff(\$datetime2);	
16	IF (difference->days <= 7) THEN	11
17	inisialisasi variabel nama =	12
	key[nama_klien]	
18	mencetak pesan notifikasi <i>deadline</i>	13
19	END IF	14
20	END IF	15
21	END FOREACH	16
22	IF (res == '') THEN	17
23	mencetak pesan tidak ada <i>deadline</i>	18
24	ELSE THEN	19
25	mencetak variabel res	
26	END IF	20
27	END IF	

2. Basic Path Testing

a) Flow Graph



Gambar 6.2 Flow graph fungsi work



b) *Cyclomatic Complexity V(G)*

- I. $V(G) = R = 7$
- II. $V(G) = E - N + 2 = 25 - 20 + 2 = 7$
- III. $V(G) = P + 1 = 6 + 1 = 7$

c) *Independent Path*

- I. Jalur 1 = 1-2-4-5-6-7-20
- II. Jalur 2 = 1-3-4-5-6-7-20
- III. Jalur 3 = 1-2-4-5-6-8-16-17-19-20
- IV. Jalur 4 = 1-2-4-5-6-8-16-18-19-20
- V. Jalur 5 = 1-2-4-5-6-8-9-14-15-8-16-17-19-20
- VI. Jalur 6 = 1-2-4-5-6-8-9-10-11-13-14-15-8-16-17-19-20
- VII. Jalur 7 = 1-2-4-5-6-8-9-10-11-12-13-14-15-8-16-18-19-20

Tabel 6.4 Hasil pengujian unit fungsi work() kelas Helper

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1-2-4-5-6-7-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log user adalah pemilik, menginisialisasi variabel data, declare variabel res = "", dan jika data==NULL.	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Valid.
2	1-3-4-5-6-7-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log user adalah Pegawai, menginisialisasi variabel data, deklarasi variabel res = "", dan jika data==NULL.	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Valid.
3	1-2-4-5-6-8-16-17-19-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log user adalah pemilik, menginisialisasi variabel data, deklarasi variabel res = "", jika data!=NULL, melakukan looping untuk mengelompokkan data yang akan dimasukkan ke variabel res, dan jika variabel, res == "".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Valid.
4	1-2-4-5-6-8-16-18-19-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak	Valid.



Tabel 6.4 Hasil pengujian unit fungsi work() kelas Helper (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
4	1-2-4-5-6-8-16-18-19-20	user adalah pemilik, menginisialisasi variabel data, deklarasi variabel res = "", jika data!=NULL, melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel res, jika variabel, res != "	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Valid.
5	1-2-4-5-6-8-9-14-15-8-16-17-19-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log user adalah pemilik, menginisialisasi variabel data, deklarasi variabel res = "", jika data!=NULL, melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel res, jika tgl_selesai>date(Y-m-d), dan jika res==".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Valid.
6	1-2-4-5-6-8-9-10-11-13-14-15-8-16-17-19-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log user adalah pemilik, menginisialisasi variabel data, menginisialisasi variabel res = "", jika data!=NULL, melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel res, jika tgl_selesai>date(Y-m-d), menghitung variabel difference, jika difference<=7, dan jika res==".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Mencetak pesan "Tidak ada <i>deadline</i> ".	Valid.
7	1-2-4-5-6-8-9-10-11-12-13-14-15-8-16-18-19-20	Kelas <i>Driver</i> memanggil fungsi work() dengan log user adalah pemilik, menginisialisasi variabel data, deklarasi variabel res = "", jika data!=NULL, melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel res, jika tgl_selesai>date(Y-	Mencetak pesan pesan <i>deadline</i> ,	Mencetak pesan pesan <i>deadline</i> .	Valid.



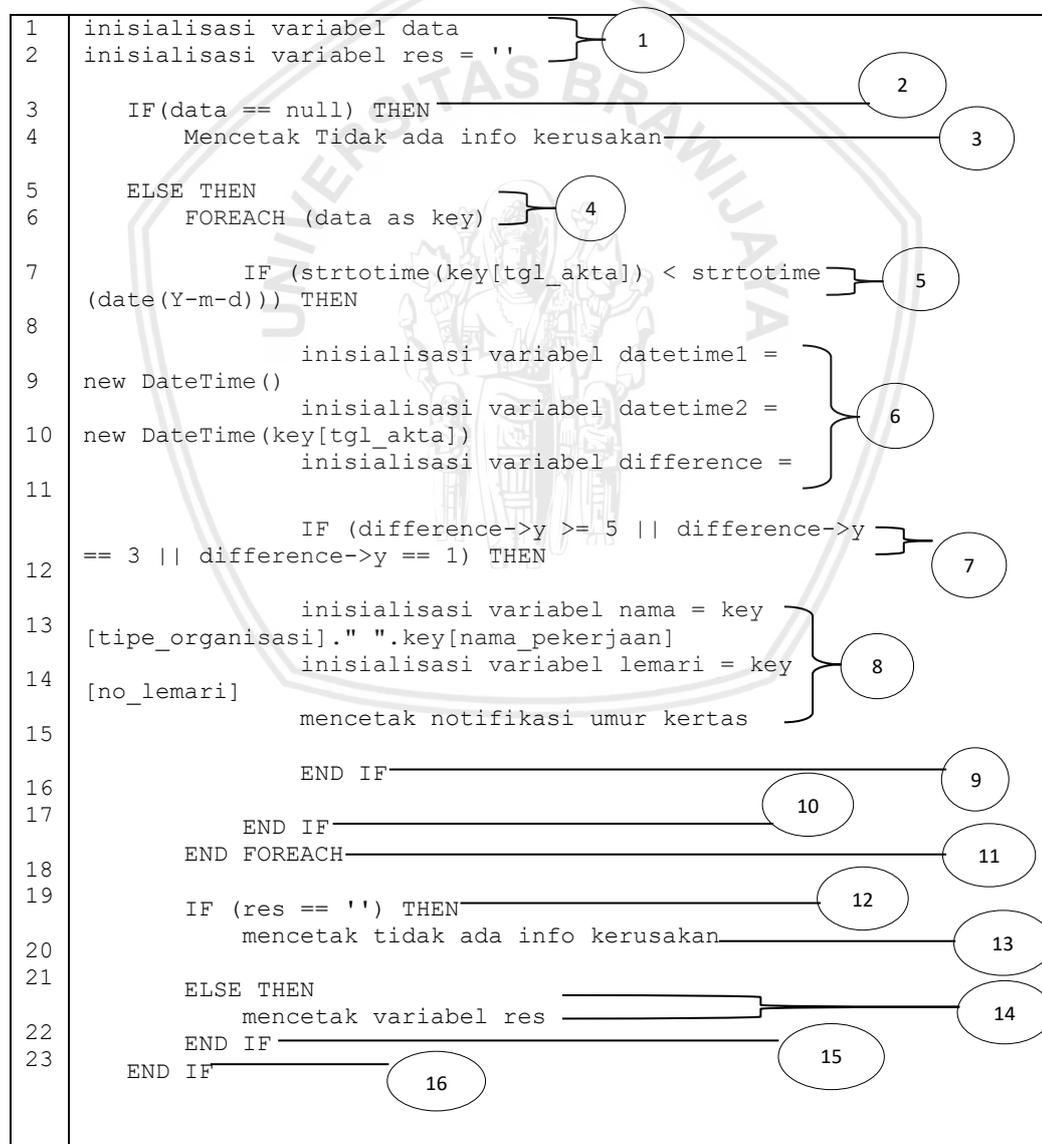
Tabel 6.4 Hasil pengujian unit fungsi work() kelas Helper (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
7	1-2-4-5-6-8-9-10-11-12-13-14-15-8-16-18-19-20	m-d), menghitung variabel difference, jika difference<=7, menambahkan data yang telah diseleksi ke dalam variabel res, dan jika res !=''.	Mencetak pesan <i>deadline</i> ,	Mencetak pesan <i>deadline</i> .	Valid.

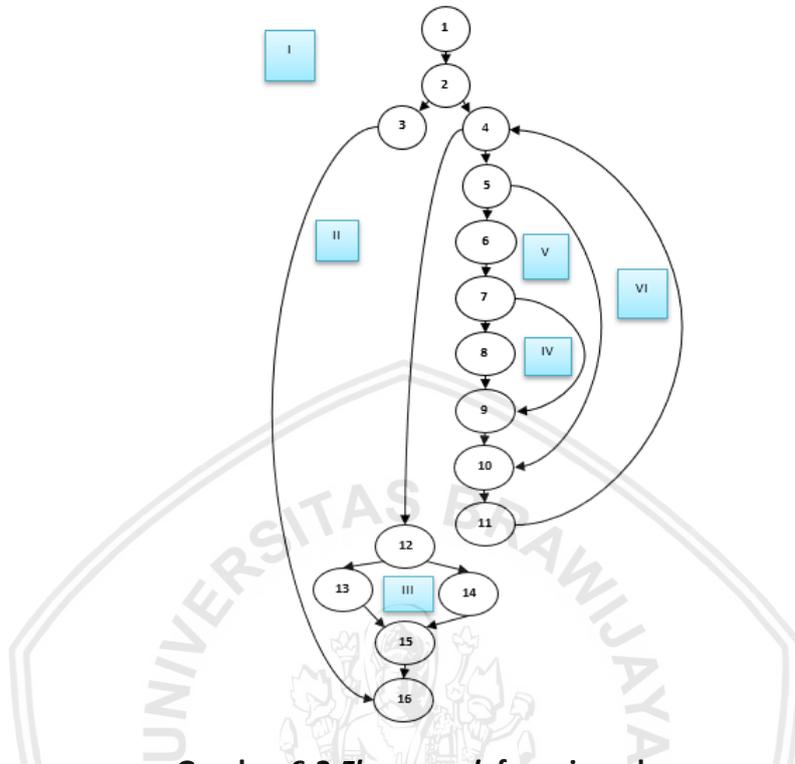
6.1.3 Pengujian Unit Kelas Helper pada Fungsi Paper()

1. Pseudocode

Tabel 6.5 Pseudocode paper



2. Basic Path Testing
 a) Flow Graph



Gambar 6.3 Flow graph fungsi work

b) Cyclomatic Complexity $V(G)$

- I. $V(G) = R = 6$
- II. $V(G) = E - N + 2 = 20 - 16 + 2 = 6$
- III. $V(G) = P + 1 = 5 + 1 = 6$

c) Independent Path

- I. Jalur 1 = 1-2-3-16
- II. Jalur 2 = 1-2-4-12-13-15-16
- III. Jalur 3 = 1-2-4-12-14-15-16
- IV. Jalur 4 = 1-2-4-5-10-11-4-12-13-15-16.
- V. Jalur 5 = 1-2-4-5-6-7-9-10-11-4-12-13-15-16
- VI. Jalur 6 = 1-2-4-5-6-7-8-9-10-11-4-12-14-15-16

Tabel 6.6 Hasil pengujian unit fungsi paper() kelas Helper

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
----	-----------	--------------	-----------------	--------	--------

Tabel 6.6 Hasil pengujian unit fungsi paper() kelas Helper (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1-2-3-16	Kelas <i>Driver</i> memanggil fungsi <code>paper()</code> dengan menginisialisasi variabel <code>data</code> dan <code>res</code> , dan jika <code>data==NULL</code> .	Mencetak "Tidak ada info kerusakan".	Mencetak "Tidak ada info kerusakan".	Valid.
2	1-2-4-12-13-15-16	Kelas <i>Driver</i> memanggil fungsi <code>paper()</code> dengan menginisialisasi variabel <code>data</code> dan <code>res</code> , jika <code>data!=NULL</code> , melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel <code>res</code> , dan jika variabel <code>res=="</code> .	Mencetak "Tidak ada info kerusakan".	Mencetak "Tidak ada info kerusakan".	Valid.
3	1-2-4-12-14-15-16	Kelas <i>Driver</i> memanggil fungsi <code>paper()</code> dengan menginisialisasi variabel <code>data</code> dan <code>res</code> , jika <code>data!=NULL</code> , melakukan looping untuk mengelompokkan data yang akan dimasukan ke variabel <code>res</code> , dan jika variabel <code>res!="</code> .	Mencetak "Tidak ada info kerusakan".	Mencetak "Tidak ada info kerusakan".	Valid.
4	1-2-4-5-10-11-4-12-13-15-16	Kelas <i>Driver</i> memanggil fungsi <code>paper()</code> dengan menginisialisasi variabel <code>data</code> dan <code>res</code> , jika <code>data!=NULL</code> , melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel <code>res</code> , jika tidak <code>key[tgl_akta] < (date(Y-m-d))</code> , dan jika variabel <code>res=="</code> .	Mencetak "Tidak ada info kerusakan".	Mencetak "Tidak ada info kerusakan".	Valid.
5	1-2-4-5-6-7-9-10-11-4-12-13-15-16	Kelas <i>Driver</i> memanggil fungsi <code>paper()</code> dengan menginisialisasi variabel <code>data</code> dan <code>res</code> , jika <code>data!=NULL</code> , melakukan looping untuk mengelompokkan data yg akan dimasukan ke variabel <code>res</code> , dan jika <code>key[tgl_akta] <</code>	Mencetak "Tidak ada info kerusakan".	Mencetak "Tidak ada info kerusakan".	Valid.



Tabel 6.6 Hasil pengujian unit fungsi paper() kelas Helper (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
5	1-2-4-5-6-7-9-10-11-4-12-13-15-16	< (date(Y-m-d)), menghitung variabel difference, jika tidak difference->y >= 5 difference->y == 3 difference->y == 1, dan jika variabel res== '.	Mencetak "Tidak ada info kerusakan".	Mencetak "Tidak ada info kerusakan".	Valid.
6	1-2-4-5-6-7-8-9-10-11-4-12-14-15-16	Kelas <i>Driver</i> memanggil fungsi <i>paper()</i> dengan menginisialisasi variabel data dan res, jika data!=NULL, melakukan looping untuk mengelompokan data yg akan dimasukan ke variabel res, jika key[tgl_akta] < (date(Y-m-d)), menghitung variabel difference, jika difference->y >= 5 difference->y == 3 difference->y == 1, mencetak pesan notifikasi, jika variabel res!= "", dan mencetak list data.	Mencetak pesan notifikasi umur kertas.	Mencetak pesan notifikasi umur kertas.	Valid.

6.2 Pengujian Integrasi

Pengujian integrasi bertujuan untuk memeriksa gabungan komponen yang sama-sama terikat pada modul yang dijalankan. Pengujian integrasi dilakukan dengan cara menguji fungsi *tambahOrder()* pada kelas *Order*. Didalam kelas *Order* akan memanggil fungsi *getKlien()*, *getInstansi()*, dan *daftar()*. Pengujian integrasi *tambahOrder()* akan ditunjukkan pada Tabel 6.7.

1. Pseudocode

Tabel 6.7 Pseudocode *tambahOrder*

1	IF (user[log] != Pin) THEN	1
2	redirect(base_url(order/pesan), refresh)	2
3	ELSE THEN	3
4	Inisialisasi variabel data[mode] = user[log]	
5	Inisialisasi variabel data[klien] = getKlien()	
6	Inisialisasi variabel data[instansi] = getInstansi()	
7	Inisialisasi variabel data[username] = user[nama]	
8	form_validation -> set_rules(klien, nama klien, required)	



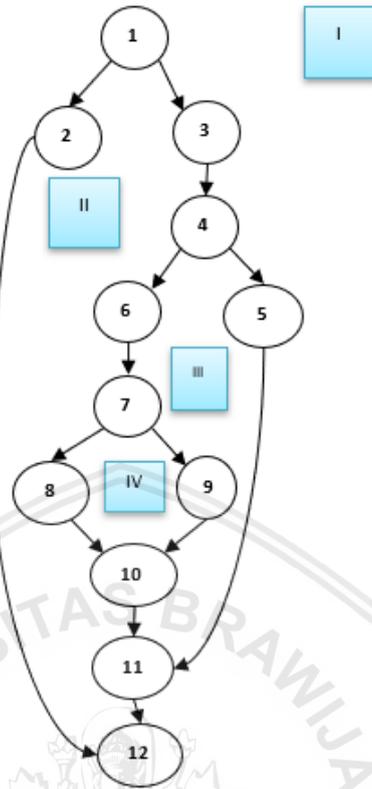
Tabel 6.7 Pseudocode tambahOrder (lanjutan)

1	IF (user[log] != Pin) THEN	1
2	redirect (base_url (order/pesan), refresh)	2
3	ELSE THEN	
4	Inisialisasi variabel data[mode] = user[log]	3
5	Inisialisasi variabel data[klien] = getKlien()	
6	Inisialisasi variabel data[instansi] = getInstansi()	
7	Inisialisasi variabel data[username] = user[nama]	
8	form_validation -> set_rules(klien, nama klien, required)	
9		
10	IF (form_validation === FALSE) THEN	4
11	Menampilkan halaman header	5
12	Menampilkan halaman nav	
13	Menampilkan halaman buat order	
14	Menampilkan halaman footer	
15	ELSE THEN	
16	Inisialisasi[upload_path] = ./dokumen/order	6
17	Inisialisasi[allowed_types] = pdf	
18	Inisialisasi[max_size] = 1024 * 12	
19	Memasukan library upload 6	
20	IF (!upload->do_upload(dokumen)) THEN	7
21	tampilkan pesan error	8
22	ELSE THEN 9	
23	Inisialisasi variabel data = data()	9
24	memasukan data ke dalam database	
25	redirect (base_url (order/pesan), refresh)	
26	END IF	10
27	END IF	11
	END IF	11

2. Basic Path Testing

a) Flow Graph





Gambar 6.4 Flow graph fungsi tambahOrder

b) Cyclomatic Complexity $V(G)$

- I. $V(G) = R = 4$
- II. $V(G) = E - N + 2 = 14 - 12 + 2 = 4$
- III. $V(G) = P + 1 = 3 + 1 = 4$

c) Independent Path

- I. Jalur 1 = 1-2-12
- II. Jalur 2 = 1-3-4-5-11-12
- III. Jalur 3 = 1-3-4-6-7-8-10-11-12
- IV. Jalur 4 = 1-3-4-6-7-9-10-11-12

Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1-2-12	Kelas Driver memanggil fungsi tambahOrder() kemudian melakukan seleksi kondisi jika user['log'] != 'Pin' maka akan menjalankan fungsi redirect ke url order/pesan.	Mengarahkan ke controller order dengan fungsi pesan, yaitu halaman order.	Mengarahkan ke controller order dengan fungsi pesan, yaitu halaman order.	Valid.

Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
2	1-3-4-5-11-12	Kelas Driver memanggil fungsi tambahOrder yang memanggil fungsi getInstance() dari kelas klien, kemudian melakukan seleksi kondisi jika user['log'] != 'Pin' dan bernilai FALSE maka akan mengambil data log variabel user yang disimpan ke dalam \$data['mode'], mengambil data klien dari model klien dan fungsi getClient ke dalam \$data['klien'], mengambil data instansi dari model order dan fungsi getInstance ke dalam \$data['instansi'], mengambil data nama variabel user yang disimpan ke dalam \$data['username'], menjalankan seleksi kondisi apabila form_validation bernilai FALSE maka akan mengirim variabel data yang diambil tadi ke dalam halaman v_buatOrder.	Menampilkan halaman tambah order dengan list klien yang ingin dibuatkan order.	Menampilkan halaman tambah order dengan list klien yang ingin dibuatkan order.	Valid.
3	1-3-4-6-7-8-10-11-12	Kelas Driver memanggil fungsi tambahOrder yang memanggil fungsi getInstance(), kemudian melakukan seleksi kondisi jika user['log'] != 'Pin' dan bernilai FALSE maka akan mengambil data log variabel user yang disimpan ke dalam \$data['mode'], mengambil data klien dari model klien dan fungsi getInstance ke dalam \$data['instansi'], mengambil data nama	Mencetak pesan gagal upload.	Mencetak pesan gagal upload.	Valid.

Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
3	1-3-4-6-7-8-10-11-12	variabel user yang disimpan ke dalam data['username'], menjalankan seleksi kondisi apabila form_validation bernilai TRUE maka akan melakukan konfigurasi <i>upload</i> . Kemudian melakukan seleksi kondisi ketika menjalan fungsi do_upload() dengan parameter dokumen bernilai FALSE maka akan membuat <i>flash data message</i> dengan nisi "Gagal Upload Dokumen". Kemudian fungsi redirect ke url order/pesan.	Mencetak pesan gagal upload.	Mencetak pesan gagal upload.	Valid.
4	1-3-4-6-7-9-10-11-12	Kelas Driver memanggil fungsi tambahOrder yang memanggil fungsi getInstance(), kemudian melakukan seleksi kondisi jika user['log'] != 'Pin' dan bernilai FALSE maka akan mengambil data log variabel user yang disimpan ke dalam \$data['mode'], mengambil data klien dari model klien dan fungsi getClient ke dalam \$data['klien'], mengambil data instansi dari model order dan fungsi getInstance ke dalam \$data['instansi'], mengambil data nama variabel user yang disimpan ke dalam \$data['username'], menjalankan seleksi kondisi apabila form_validation bernilai TRUE maka akan melakukan konfigurasi <i>upload</i> .	Mencetak pesan "Berhasil menambahkan Order" dan data dimasukan ke dalam <i>database</i> .	Mencetak pesan "Berhasil menambahkan Order" dan data dimasukan ke dalam <i>database</i> .	Valid.

Tabel 6.8 Hasil pengujian integrasi fungsi tambahOrder() kelas Order (lanjutan)

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
4	1-3-4-6-7-9-10-11-12	Kemudian melakukan seleksi kondisi ketika menjalan fungsi do_upload() dengan parameter dokumen bernilai TRUE maka akan mengambil data dari proses unggah, kemudian memasukan dalam <i>database</i> . melalui model order dan fungsi daftar dengan parameter \$data['file_name']. Dan membuat <i>flash data message</i> dengan isi "Berhasil Upload Dokumen". Kemudian fungsi redirect ke url order/pesan.	Mencetak pesan "Berhasil menambahkan Order" dan data dimasukan ke dalam <i>database</i> .	Mencetak pesan "Berhasil menambahkan Order" dan data dimasukan ke dalam <i>database</i> .	Valid.

6.3 Pengujian Validasi

Pengujian validasi dijalankan selepas tahapan sebelumnya, yakni implementasi tuntas. Pengujian validasi menggunakan pengujian *blackbox* yang dijalankan untuk mengetahui sistem yang dibentuk telah memenuhi seluruh kebutuhan yang dibutuhkan serta mengetahui apakah seluruh fungsi sudah memenuhi kebutuhan. Pengujian *blackbox* digunakan untuk memastikan masukan sistem sesuai tanpa mengetahui kode program. Pengujian validasi ditunjukkan antara pada Tabel 6.9 dan Tabel 6.101.

6.3.1 Pengujian Validasi Login

1. Kasus uji berhasil *login*.

Tabel 6.9 Kasus uji berhasil login

Nama Kasus Uji	Kasus uji berhasil melakukan <i>login</i> .
Prosedur uji	<ol style="list-style-type: none"> 1. Mengisi kolom username. 2. Mengisi kolom password. 3. Menekan tombol <i>login</i>.
Hasil yang diharapkan	Mempresentasikan halaman <i>dashboard</i> user.
Hasil uji	Mempresentasikan halaman <i>dashboard</i> user.

Tabel 6.9 Kasus uji berhasil melakukan login (lanjutan)

Nama Kasus Uji	Kasus uji berhasil melakukan <i>login</i> .
Status	Valid.

2. Kasus uji gagal melakukan *login* dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.10 Kasus uji gagal melakukan login dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal melakukan <i>login</i> dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Mengisi kolom <i>username</i> . 2. Tidak mengisi kolom <i>password</i> . 3. Menekan tombol <i>login</i> .
Hasil yang diharapkan	Mencetak pesan "Please fill out this field" pada kolom <i>password</i> .
Hasil uji	Mencetak pesan "Please fill out this field" pada kolom <i>password</i> .
Status	Valid.

3. Kasus uji gagal melakukan *login* dengan masukan yang tidak sesuai pada kolom *username* atau *password*.

Tabel 6.11 Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom username atau password

Nama Kasus Uji	Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom <i>username</i> atau <i>password</i> ..
Prosedur uji	1. Mengisi kolom <i>username</i> atau <i>password</i> dengan nilai yang tidak sesuai. 2. Menekan tombol <i>login</i> .
Hasil yang diharapkan	Mencetak pesan "Username atau Password Salah".
Hasil uji	Mencetak pesan "Username atau Password Salah".
Status	Valid.

6.3.2 Pengujian Validasi Logout

1. Kasus uji berhasil logout.

Tabel 6.12 Kasus uji berhasil logout

Nama Kasus Uji	Kasus uji berhasil logout.
-----------------------	----------------------------

Tabel 6.12 Kasus uji berhasil logout (lanjutan)

Nama Kasus Uji	Kasus uji berhasil logout.
Prosedur uji	1. Memilih tombol nama. 2. Memilih tombol logout.
Hasil yang diharapkan	Mempresentasikan halaman login.
Hasil uji	Mempresentasikan halaman login.
Status	Valid.

6.3.3 Pengujian Validasi Melihat Akun Pegawai

1. Kasus uji berhasil melihat akun pegawai.

Tabel 6.13 Kasus uji berhasil melihat akun pegawai

Nama Kasus Uji	Kasus uji berhasil melihat akun pegawai.
Prosedur uji	1. Login sebagai pemilik. 2. Memilih menu akun pegawai pegawai.
Hasil yang diharapkan	Mempresentasikan data akun pegawai.
Hasil uji	Mempresentasikan data akun pegawai.
Status	Valid.

6.3.4 Pengujian Validasi Menambah Akun Pegawai

1. Kasus uji berhasil menambah akun pegawai.

Tabel 6.14 Kasus uji berhasil menambah akun pegawai

Nama Kasus Uji	Kasus uji berhasil menambah akun pegawai.
Prosedur uji	1. Mengisi form tambah akun pegawai. 2. Memilih tombol tambah pegawai.
Hasil yang diharapkan	Mencetak pesan "Pendaftaran berhasil".
Hasil uji	Mencetak pesan "Pendaftaran berhasil".
Status	Valid.

2. Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.15 Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem.
-----------------------	--

Tabel 6.15 Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem (lanjutan)

Nama Kasus Uji	Kasus uji gagal menambah akun dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form menambah akun pegawai dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

3. Kasus uji gagal menambah akun dengan *username* yang sudah terdapat pada daftar akun.

Tabel 6.16 Kasus uji gagal menambah akun dengan username yang sudah terdapat pada daftar akun

Nama Kasus Uji	Kasus uji gagal menambah akun dengan <i>username</i> yang sudah terdapat pada daftar akun.
Prosedur uji	1. Mengisi form tambah akun. 2. Mengisi kolom <i>username</i> yang sudah terdapat pada daftar akun. 3. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan "Username telah digunakan".
Hasil uji	Mencetak pesan "Username telah digunakan".
Status	Valid.

6.3.5 Pengujian Validasi Mengubah Akun Pegawai

1. Kasus uji berhasil mengubah akun pegawai.

Tabel 6.17 Kasus uji berhasil mengubah akun pegawai

Nama Kasus Uji	Kasus uji berhasil melihat akun pegawai.
Prosedur uji	1. Mengubah data akun pegawai. 2. Memilih tombol ubah.
Hasil yang diharapkan	Akun pegawai berhasil diubah.
Hasil uji	Akun pegawai berhasil diubah.
Status	Valid.

2. Kasus uji gagal mengubah akun pegawai dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.18 Kasus uji gagal mengubah akun pegawai dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah akun pegawai dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Mengubah data akun pegawai dengan mengosongkan data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.6 Pengujian Validasi Menghapus Akun Pegawai

1. Kasus uji berhasil menghapus akun pegawai.

Tabel 6.19 Kasus uji berhasil menghapus akun pegawai

Nama Kasus Uji	Kasus uji berhasil menghapus akun pegawai.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih tombol hapus pada akun pegawai yang dipilih. 2. Memilih tombol hapus. 3. memilih tombol "Ya, Hapus!".
Hasil yang diharapkan	Akun pegawai berhasil dihapus.
Hasil uji	Akun pegawai berhasil dihapus.
Status	Valid.

2. Kasus uji gagal menghapus akun pegawai.

Tabel 6.20 Kasus uji gagal menghapus akun pegawai

Nama Kasus Uji	Kasus uji gagal menghapus akun pegawai.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih tombol hapus pada akun pegawai yang dipilih. 2. Memilih tombol hapus. 3. memilih tombol "Cancel".
Hasil yang diharapkan	Akun pegawai yang dipilih tidak dihapus.
Hasil uji	Akun pegawai yang dipilih tidak dihapus.
Status	Valid.

6.3.7 Pengujian Validasi Menambah Klien

1. Kasus uji berhasil menambah klien.

Tabel 6.21 Kasus uji berhasil menambah klien

Nama Kasus Uji	Kasus uji berhasil menambah klien.
Prosedur uji	1. Mengisi form tambah klien. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Pendaftaran berhasil”.
Hasil uji	Mencetak pesan “Pendaftaran berhasil”.
Status	Valid.

2. Kasus uji gagal menambah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.22 Kasus uji gagal menambah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem..
Prosedur uji	1. Tidak mengisi form tambah klien. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Please fill out this field”.
Hasil uji	Mencetak pesan “Please fill out this field”.
Status	Valid.

3. Kasus uji gagal menambah klien dengan nomor KTP yang sudah terdapat pada daftar klien.

Tabel 6.23 Kasus uji gagal menambah klien dengan nomor KTP yang sudah terdapat pada daftar klien

Nama Kasus Uji	Kasus uji gagal menambah klien dengan nomor KTP yang sudah terdapat pada daftar klien.
Prosedur uji	1. Mengisi form tambah klien. 2. Mengisi kolom nomor KTP yang sudah terdapat pada daftar akun. 3. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Nomor KTP telah terdaftar”.
Hasil uji	Mencetak pesan “Nomor KTP telah terdaftar”.
Status	Valid.

6.3.8 Pengujian Validasi Melihat Klien

1. Kasus uji berhasil melihat klien.

Tabel 6.24 Kasus uji berhasil melihat klien

Nama Kasus Uji	Kasus uji berhasil melihat klien.
Prosedur uji	1. Memilih menu klien.
Hasil yang diharapkan	Mempresentasikan data klien.
Hasil uji	Mempresentasikan data klien.
Status	Valid.

6.3.9 Pengujian Validasi Mengubah Klien

1. Kasus uji berhasil mengubah klien.

Tabel 6.25 Kasus uji berhasil mengubah klien

Nama Kasus Uji	Kasus uji berhasil mengubah klien.
Prosedur uji	1. Mengubah data klien. 2. Memilih tombol ubah.
Hasil yang diharapkan	Data klien berhasil diubah.
Hasil uji	Data klien berhasil diubah.
Status	Valid.

2. Kasus uji gagal mengubah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.26 Kasus uji gagal mengubah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah klien dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Mengubah data akun pegawai dengan mengosongkan data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.10 Pengujian Validasi Melihat Riwayat Klien

1. Kasus uji berhasil melihat riwayat klien.

Tabel 6.27 Kasus uji berhasil melihat riwayat klien

Nama Kasus Uji	Kasus uji berhasil melihat riwayat klien.
Prosedur uji	1. Memilih menu klien. 2. Memilih tombol riwayat klien.
Hasil yang diharapkan	Mempresentasikan data riwayat klien yang terdapat rincian riwayat order klien.
Hasil uji	Mempresentasikan data riwayat klien yang terdapat rincian riwayat order klien.
Status	Valid.

6.3.11 Pengujian Validasi Melihat Pesan Nama Organisasi

1. Kasus uji berhasil melihat pesan nama organisasi.

Tabel 6.28 Kasus uji berhasil melihat pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil melihat pesan nama organisasi.
Prosedur uji	1. Memilih menu organisasi. 2. Memilih sub-menu pesan nama.
Hasil yang diharapkan	Mempresentasikan data pesan nama organisasi.
Hasil uji	Mempresentasikan data pesan nama organisasi.
Status	Valid.

6.3.12 Pengujian Validasi Menambah Pesan Nama Organisasi

1. Kasus uji berhasil menambah pesan nama organisasi.

Tabel 6.29 Kasus uji berhasil menambah pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil pesan nama organisasi.
Prosedur uji	1. Mengisi form pesan nama organisasi. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan "Order berhasil ditambah".
Hasil uji	Mencetak pesan "Order berhasil ditambah".
Status	Valid.

2. Kasus uji gagal menambah pesan nama organisasi dengan *file* yang diunggah bukan ekstensi .pdf.

Tabel 6.30 Kasus uji gagal menambah pesan nama organisasi dengan file yang diunggah bukan ekstensi .pdf

Nama Kasus Uji	Kasus uji gagal menambah pesan nama organisasi dengan <i>file</i> yang diunggah bukan ekstensi .pdf.
Prosedur uji	<ol style="list-style-type: none"> 1. Tidak mengisi form pesan nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Berkas gagal diunggah”.
Hasil uji	Mencetak pesan “Berkas gagal diunggah”.
Status	Valid.

3. Kasus uji gagal menambah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.31 Kasus uji gagal menambah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Tidak mengisi form pesan nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Please fill out this field”.
Hasil uji	Mencetak pesan “Please fill out this field”.
Status	Valid.

6.3.13 Pengujian Validasi Melihat Rincian Berkas Klien pada Pesan Nama Organisasi

1. Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi.

Tabel 6.32 Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu organisasi. 2. Memilih sub-menu pesan nama. 3. Memilih klien. 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada pesan nama organisasi.

Tabel 6.32 Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi (lanjutan)

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada pesan nama organisasi.
Hasil uji	Mempresentasikan rincian berkas klien pada pesan nama organisasi.
Status	Valid.

6.3.14 Pengujian Validasi Mengubah Pesan Nama Organisasi

1. Kasus uji berhasil mengubah pesan nama organisasi.

Tabel 6.33 Kasus uji berhasil mengubah pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil mengubah pesan nama organisasi.
Prosedur uji	1. Mengisi form ubah pesan nama organisasi. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Order berhasil diubah".
Hasil uji	Mencetak pesan "Order berhasil diubah".
Status	Valid.

2. Kasus uji gagal mengubah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.34 Kasus uji gagal mengubah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form ubah pesan nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.15 Pengujian Validasi Menolak Pesan Nama Organisasi

1. Kasus uji berhasil menolak pesan nama organisasi.

Tabel 6.35 Kasus uji berhasil menolak pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil menolak pesan nama organisasi.
-----------------------	---

Tabel 6.35 Kasus uji berhasil menolak pesan nama organisasi (lanjutan)

Nama Kasus Uji	Kasus uji berhasil menolak pesan nama organisasi.
Prosedur uji	1. Mengisi form tolak pesan nama organisasi. 2. Memilih tombol tolak.
Hasil yang diharapkan	Order berhasil ditolak.
Hasil uji	Order berhasil ditolak.
Status	Valid.

2. Kasus uji gagal menolak pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.36 Kasus uji gagal menolak pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menolak pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form tolak pesan nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tolak.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.16 Pengujian Validasi Menghapus Pesan Nama Organisasi

1. Kasus uji berhasil menghapus pesan nama organisasi.

Tabel 6.37 Kasus uji berhasil menghapus pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil menghapus akun pegawai.
Prosedur uji	1. Memilih tombol hapus pada akun pegawai yang dipilih.
Hasil yang diharapkan	Data pesan nama berhasil dihapus.
Hasil uji	Data pesan nama berhasil dihapus.
Status	Valid.

6.3.17 Pengujian Validasi Memverifikasi Pesan Nama Organisasi

1. Kasus uji berhasil memverifikasi pesan nama organisasi.

Tabel 6.38 Kasus uji berhasil memverifikasi pesan nama organisasi

Nama Kasus Uji	Kasus uji berhasil memverifikasi pesan nama organisasi.
Prosedur uji	1. Memilih pesan nama organisasi yang ingin diverifikasi. 2. Memilih tombol verifikasi pesan nama organisasi.
Hasil yang diharapkan	Mencetak pesan “Order berhasil terverifikasi”.
Hasil uji	Mencetak pesan “Order berhasil terverifikasi”.
Status	Valid.

6.3.18 Pengujian Validasi Melihat Buat Akta Organisasi

1. Kasus uji berhasil melihat buat akta organisasi.

Tabel 6.39 Kasus uji berhasil melihat buat akta organisasi

Nama Kasus Uji	Kasus uji berhasil melihat buat akta organisasi.
Prosedur uji	1. Memilih menu organisasi. 2. Memilih sub-menu buat akta.
Hasil yang diharapkan	Mempresentasikan data buat akta organisasi.
Hasil uji	Mempresentasikan data buat akta organisasi.
Status	Valid.

6.3.19 Pengujian Validasi Melihat Rincian Berkas Klien pada Buat Akta Organisasi

1. Kasus uji berhasil melihat rincian berkas klien pada buat akta organisasi.

Tabel 6.40 Kasus uji berhasil melihat rincian berkas klien pada buat akta organisasi

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada buat akta organisasi.
Prosedur uji	1. Memilih menu organisasi. 2. Memilih sub-menu buat akta. 3. Memilih klien. 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada buat akta organisasi.
Hasil uji	Mempresentasikan rincian berkas klien pada buat akta organisasi.
Status	Valid.



6.3.20 Pengujian Validasi Membarui Status Buat Akta Organisasi

1. Kasus uji berhasil membarui status buat akta organisasi.

Tabel 6.41 Kasus uji berhasil membarui status buat akta organisasi

Nama Kasus Uji	Kasus uji berhasil membarui status buat akta organisasi.
Prosedur uji	1. Memilih data yang statusnya akan diperbarui. 2. Membarui status buat akta organisasi.
Hasil yang diharapkan	Status buat akta organisasi berhasil diperbarui.
Hasil uji	Status buat akta organisasi berhasil diperbarui.
Status	Valid.

6.3.21 Pengujian Validasi Mengonfirmasi Status Buat Akta Organisasi Selesai

1. Kasus uji berhasil membarui status buat akta organisasi selesai.

Tabel 6.42 Kasus uji berhasil membarui status buat akta organisasi selesai

Nama Kasus Uji	Kasus uji berhasil membarui status buat akta organisasi selesai.
Prosedur uji	1. Memilih data yang statusnya akan dikonfirmasi selesai. 2. mengonfirmasi status buat akta organisasi selesai.
Hasil yang diharapkan	Status buat akta organisasi berhasil dikonfirmasi selesai.
Hasil uji	Status buat akta organisasi berhasil dikonfirmasi selesai.
Status	Valid.

6.3.22 Pengujian Validasi Melihat Pendaftaran SK Organisasi

1. Kasus uji berhasil melihat pendaftaran SK organisasi.

Tabel 6.43 Kasus uji berhasil melihat pendaftaran SK organisasi

Nama Kasus Uji	Kasus uji berhasil melihat pendaftaran SK organisasi.
Prosedur uji	1. Memilih menu organisasi. 2. Memilih sub-menu pendaftaran SK.
Hasil yang diharapkan	Mempresentasikan data pendaftaran SK organisasi.
Hasil uji	Mempresentasikan data pendaftaran SK organisasi.
Status	Valid.

6.3.23 Pengujian Validasi Mengunggah Berkas SK Organisasi

1. Kasus uji berhasil mengunggah berkas SK organisasi.

Tabel 6.44 Kasus uji berhasil mengunggah berkas SK organisasi

Nama Kasus Uji	Kasus uji berhasil mengunggah berkas SK organisasi.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu organisasi. 2. Memilih sub-menu pendaftaran SK. 3. Memilih data berkas SK. 4. Memilih tombol unggah berkas SK. 5. Memilih menu <i>choose file</i>. 6. Memilih berkas yang ingin diunggah. 7. Memilih tombol <i>open</i>. 8. Memilih tombol <i>upload</i>.
Hasil yang diharapkan	Berkas SK organisasi berhasil diunggah.
Hasil uji	Berkas SK organisasi berhasil diunggah.
Status	Valid.

2. Kasus uji gagal mengunggah berkas SK organisasi dengan *file* yang diunggah bukan ekstensi .pdf.

Tabel 6.45 Kasus uji berhasil mengunggah berkas SK organisasi dengan file yang diunggah bukan ekstensi .pdf.

Nama Kasus Uji	Kasus uji gagal mengunggah berkas SK organisasi dengan <i>file</i> yang diunggah bukan ekstensi .pdf.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu organisasi. 2. Memilih sub-menu pendaftaran SK. 3. Memilih data berkas SK. 4. Memilih tombol unggah berkas SK. 5. Memilih menu <i>choose file</i>. 6. Memilih berkas yang ingin diunggah bukan dengan ekstensi <i>file</i> .pdf. 7. Memilih tombol <i>open</i>. 8. Memilih tombol <i>upload</i>.
Hasil yang diharapkan	Berkas SK organisasi gagal diunggah.
Hasil uji	Berkas SK organisasi gagal diunggah.
Status	Valid.

6.3.24 Pengujian Validasi Melihat Rincian Berkas Klien pada Pendaftaran SK Organisasi

1. Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK organisasi.

Tabel 6.46 Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK organisasi

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK organisasi.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu organisasi. 2. Memilih sub-menu pendaftaran SK. 3. Memilih klien. 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada pendaftaran SK organisasi.
Hasil uji	Mempresentasikan rincian berkas klien pada pendaftaran SK organisasi.
Status	Valid.

6.3.25 Pengujian Validasi Melihat Cek nama Wasiat

1. Kasus uji berhasil melihat cek nama wasiat.

Tabel 6.47 Kasus uji berhasil melihat cek nama wasiat

Nama Kasus Uji	Kasus uji berhasil melihat cek nama wasiat.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu wasiat. 2. Memilih sub-menu cek nama.
Hasil yang diharapkan	Mempresentasikan data cek nama wasiat.
Hasil uji	Mempresentasikan data cek nama wasiat.
Status	Valid.

6.3.26 Pengujian Validasi Menambah Cek nama Wasiat

1. Kasus uji berhasil menambah cek nama wasiat.

Tabel 6.48 Kasus uji berhasil menambah cek nama wasiat

Nama Kasus Uji	Kasus uji berhasil cek nama wasiat.
Prosedur uji	<ol style="list-style-type: none"> 1. Mengisi form cek nama wasiat. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan "Order berhasil ditambah".
Hasil uji	Mencetak pesan "Order berhasil ditambah".
Status	Valid.

2. Kasus uji gagal menambah cek nama wasiat dengan *file* yang diunggah bukan ekstensi .pdf.

Tabel 6.49 Kasus uji gagal menambah cek nama wasiat dengan file yang diunggah bukan ekstensi .pdf

Nama Kasus Uji	Kasus uji gagal menambah cek nama wasiat dengan <i>file</i> yang diunggah bukan ekstensi .pdf.
Prosedur uji	1. Tidak mengisi form cek nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Berkas gagal diunggah”.
Hasil uji	Mencetak pesan “Berkas gagal diunggah”.
Status	Valid.

3. Kasus uji gagal menambah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.50 Kasus uji gagal menambah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form cek nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Please fill out this field”.
Hasil uji	Mencetak pesan “Please fill out this field”.
Status	Valid.

6.3.27 Pengujian Validasi Melihat Rincian Berkas Klien pada Cek nama Wasiat

1. Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat.

Tabel 6.51 Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat.
Prosedur uji	1. Memilih menu wasiat. 2. Memilih sub-menu cek nama. 3. Memilih klien. 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada cek nama wasiat.

Tabel 6.51 Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat (lanjutan)

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada cek nama wasiat.
Hasil uji	Mempresentasikan rincian berkas klien pada cek nama wasiat.
Status	Valid.

6.3.28 Pengujian Validasi Memverifikasi Cek Nama Wasiat

1. Kasus uji berhasil memverifikasi cek nama wasiat.

Tabel 6.52 Kasus uji berhasil memverifikasi cek nama wasiat

Nama Kasus Uji	Kasus uji berhasil memverifikasi cek nama wasiat.
Prosedur uji	1. Memilih cek nama wasiat yang ingin diverifikasi. 2. Memilih tombol verifikasi cek nama wasiat.
Hasil yang diharapkan	Mencetak pesan "Order berhasil terverifikasi".
Hasil uji	Mencetak pesan "Order berhasil terverifikasi".
Status	Valid.

6.3.29 Pengujian Validasi Menolak Cek Nama Wasiat

1. Kasus uji berhasil menolak cek nama wasiat.

Tabel 6.53 Kasus uji berhasil menolak cek nama wasiat

Nama Kasus Uji	Kasus uji berhasil menolak cek nama wasiat.
Prosedur uji	1. Mengisi form tolak cek nama wasiat. 2. Memilih tombol tolak.
Hasil yang diharapkan	Order berhasil ditolak.
Hasil uji	Order berhasil ditolak.
Status	Valid.

2. Kasus uji gagal menolak pesan nama organisasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.54 Kasus uji gagal menolak cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menolak cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form tolak cek nama dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.54 Kasus uji gagal menolak cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem (lanjutan)

Nama Kasus Uji	Kasus uji gagal menolak cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	2. Memilih tombol tolak.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.30 Pengujian Validasi Mengubah Cek nama Wasiat

1. Kasus uji berhasil mengubah cek nama wasiat.

Tabel 6.55 Kasus uji berhasil mengubah cek nama wasiat

Nama Kasus Uji	Kasus uji berhasil mengubah cek nama wasiat.
Prosedur uji	1. Mengisi form ubah cek nama wasiat. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Order berhasil diubah".
Hasil uji	Mencetak pesan "Order berhasil diubah".
Status	Valid.

2. Kasus uji gagal mengubah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.56 Kasus uji gagal mengubah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah cek nama wasiat dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form ubah cek nama dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.31 Pengujian Validasi Memverifikasi Status Pewasiat

1. Kasus uji berhasil memverifikasi status pewasiat.

Tabel 6.57 Kasus uji berhasil memverifikasi status pewasiat

Nama Kasus Uji	Kasus uji berhasil memverifikasi status pewasiat.
Prosedur uji	1. Memilih status pewasiat yang ingin diverifikasi. 2. Memilih tombol verifikasi status pewasiat.
Hasil yang diharapkan	Mencetak pesan “Pewasiat berhasil terverifikasi”.
Hasil uji	Mencetak pesan “Pewasiat berhasil terverifikasi”.
Status	Valid.

6.3.32 Pengujian Validasi Melihat Buat Akta Wasiat

1. Kasus uji berhasil melihat buat akta wasiat.

Tabel 6.58 Kasus uji berhasil melihat buat akta wasiat

Nama Kasus Uji	Kasus uji berhasil melihat buat akta wasiat.
Prosedur uji	1. Memilih menu wasiat. 2. Memilih sub-menu buat akta.
Hasil yang diharapkan	Mempresentasikan data buat akta wasiat.
Hasil uji	Mempresentasikan data buat akta wasiat.
Status	Valid.

6.3.33 Pengujian Validasi Melihat Rincian Berkas Klien pada Buat Akta Wasiat

1. Kasus uji berhasil melihat rincian berkas klien pada buat akta wasiat.

Tabel 6.59 Kasus uji berhasil melihat rincian berkas klien pada buat akta wasiat

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada buat akta wasiat.
Prosedur uji	1. Memilih menu wasiat. 2. Memilih sub-menu buat akta. 3. Memilih klien. 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada buat akta wasiat.
Hasil uji	Mempresentasikan rincian berkas klien pada buat akta wasiat.
Status	Valid.



6.3.34 Pengujian Validasi Membarui Status Buat Akta Wasiat

1. Kasus uji berhasil membarui status buat akta wasiat.

Tabel 6.60 Kasus uji berhasil membarui status buat akta wasiat

Nama Kasus Uji	Kasus uji berhasil membarui status buat akta wasiat.
Prosedur uji	1. Memilih data yang statusnya akan diperbarui. 2. Membarui status buat akta wasiat.
Hasil yang diharapkan	Status buat akta wasiat berhasil diperbarui.
Hasil uji	Status buat akta wasiat berhasil diperbarui.
Status	Valid.

6.3.35 Pengujian Validasi Mengonfirmasi Status Buat Akta Wasiat Selesai

1. Kasus uji berhasil mengonfirmasi status buat akta wasiat selesai.

Tabel 6.61 Kasus uji berhasil mengonfirmasi status buat akta wasiat selesai

Nama Kasus Uji	Kasus uji berhasil mengonfirmasi status buat akta wasiat selesai.
Prosedur uji	1. Memilih data yang statusnya akan dikonfirmasi selesai. 2. mengonfirmasi status buat akta wasiat selesai.
Hasil yang diharapkan	Status buat akta wasiat berhasil dikonfirmasi selesai.
Hasil uji	Status buat akta wasiat berhasil dikonfirmasi selesai.
Status	Valid.

6.3.36 Pengujian Validasi Melihat Pendaftaran SK Wasiat

1. Kasus uji berhasil melihat pendaftaran SK wasiat.

Tabel 6.62 Kasus uji berhasil melihat pendaftaran SK wasiat

Nama Kasus Uji	Kasus uji berhasil melihat pendaftaran SK wasiat.
Prosedur uji	1. Memilih menu wasiat. 2. Memilih sub-menu pendaftaran SK.
Hasil yang diharapkan	Mempresentasikan data pendaftaran SK wasiat.
Hasil uji	Mempresentasikan data pendaftaran SK wasiat.
Status	Valid.



6.3.37 Pengujian Validasi Mengunggah Berkas SK Wasiat

1. Kasus uji berhasil mengunggah berkas SK wasiat.

Tabel 6.63 Kasus uji berhasil mengunggah berkas SK wasiat

Nama Kasus Uji	Kasus uji berhasil mengunggah berkas SK wasiat.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu wasiat. 2. Memilih sub-menu pendaftaran SK. 3. Memilih data berkas SK. 4. Memilih tombol unggah berkas SK. 5. Memilih menu <i>choose file</i>. 6. Memilih berkas yang ingin diunggah. 7. Memilih tombol <i>open</i>. 8. Memilih tombol <i>upload</i>.
Hasil yang diharapkan	Berkas SK wasiat berhasil diunggah.
Hasil uji	Berkas SK wasiat berhasil diunggah.
Status	Valid.

2. Kasus uji gagal mengunggah berkas SK wasiat dengan *file* yang diunggah bukan ekstensi .pdf.

Tabel 6.64 Kasus uji berhasil mengunggah berkas SK wasiat dengan file yang diunggah bukan ekstensi .pdf

Nama Kasus Uji	Kasus uji gagal mengunggah berkas SK wasiat dengan <i>file</i> yang diunggah bukan ekstensi .pdf.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu wasiat. 2. Memilih sub-menu pendaftaran SK. 3. Memilih data berkas SK. 4. Memilih tombol unggah berkas SK. 5. Memilih menu <i>choose file</i>. 6. Memilih berkas yang ingin diunggah bukan dengan ekstensi <i>file</i> .pdf. 7. Memilih tombol <i>open</i>. 8. Memilih tombol <i>upload</i>.
Hasil yang diharapkan	Berkas SK Wasiat gagal diunggah.
Hasil uji	Berkas SK Wasiat gagal diunggah.
Status	Valid.

6.3.38 Pengujian Validasi Melihat Rincian Berkas Klien pada Pendaftaran SK Wasiat

1. Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK wasiat.

Tabel 6.65 Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK wasiat

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK wasiat.
Prosedur uji	<ol style="list-style-type: none">1. Memilih menu wasiat.2. Memilih sub-menu pendaftaran SK.3. Memilih klien.4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada pendaftaran SK wasiat.
Hasil uji	Mempresentasikan rincian berkas klien pada pendaftaran SK wasiat.
Status	Valid.

6.3.39 Pengujian Validasi Melihat Buat Akta Fidusia

1. Kasus uji berhasil melihat buat akta fidusia.

Tabel 6.66 Kasus uji berhasil melihat buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil melihat buat akta fidusia.
Prosedur uji	<ol style="list-style-type: none">1. Memilih menu fidusia.2. Memilih sub-menu buat akta.
Hasil yang diharapkan	Mempresentasikan data buat akta fidusia.
Hasil uji	Mempresentasikan data buat akta fidusia.
Status	Valid.

6.3.40 Pengujian Validasi Menambah Buat Akta Fidusia

1. Kasus uji berhasil menambah buat akta fidusia.

Tabel 6.67 Kasus uji berhasil menambah buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil buat akta fidusia.
Prosedur uji	<ol style="list-style-type: none">1. Mengisi form buat akta fidusia.2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan "Order berhasil ditambah".
Hasil uji	Mencetak pesan "Order berhasil ditambah".

Tabel 6.67 Kasus uji berhasil menambah buat akta fidusia (lanjutan)

Nama Kasus Uji	Kasus uji berhasil buat akta fidusia.
Status	Valid.

2. Kasus uji gagal menambah buat akta fidusia dengan *file* yang diunggah bukan ekstensi .pdf.

Tabel 6.68 Kasus uji gagal menambah buat akta fidusia dengan file yang diunggah bukan ekstensi .pdf

Nama Kasus Uji	Kasus uji gagal menambah buat akta fidusia dengan <i>file</i> yang diunggah bukan ekstensi .pdf.
Prosedur uji	1. Tidak mengisi form buat akta dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Berkas gagal diunggah”.
Hasil uji	Mencetak pesan “Berkas gagal diunggah”.
Status	Valid.

3. Kasus uji gagal menambah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.69 Kasus uji gagal menambah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form buat akta dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Please fill out this field”.
Hasil uji	Mencetak pesan “Please fill out this field”.
Status	Valid.

6.3.41 Pengujian Validasi Melihat Rincian Berkas Klien pada Buat akta Fidusia

1. Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia.

Tabel 6.70 Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia.
Prosedur uji	1. Memilih menu fidusia. 2. Memilih sub-menu buat akta.

Tabel 6.70 Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia (lanjutan)

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada buat akta fidusia.
Prosedur uji	3. Memilih Klien 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada buat akta fidusia.
Hasil uji	Mempresentasikan rincian berkas klien pada buat akta fidusia.
Status	Valid.

6.3.42 Pengujian Validasi Memverifikasi Buat Akta Fidusia

1. Kasus uji berhasil memverifikasi buat akta fidusia.

Tabel 6.71 Kasus uji berhasil memverifikasi buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil memverifikasi buat akta fidusia.
Prosedur uji	1. Memilih buat akta fidusia yang ingin diverifikasi. 2. Memilih tombol verifikasi buat akta fidusia.
Hasil yang diharapkan	Mencetak pesan "Order berhasil terverifikasi".
Hasil uji	Mencetak pesan "Order berhasil terverifikasi".
Status	Valid.

6.3.43 Pengujian Validasi Menolak Buat Akta Fidusia

1. Kasus uji berhasil menolak buat akta fidusia.

Tabel 6.72 Kasus uji berhasil menolak buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil menolak buat akta fidusia.
Prosedur uji	1. Mengisi form tolak buat akta fidusia. 2. Memilih tombol tolak.
Hasil yang diharapkan	Order berhasil ditolak.
Hasil uji	Order berhasil ditolak
Status	Valid.

2. Kasus uji gagal menolak buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.73 Kasus uji gagal menolak buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menolak buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form tolak buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tolak.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.44 Pengujian Validasi Mengubah Buat Akta Fidusia

1. Kasus uji berhasil mengubah buat akta fidusia.

Tabel 6.74 Kasus uji berhasil mengubah buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil mengubah buat akta fidusia.
Prosedur uji	1. Mengisi form ubah buat akta fidusia. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Order berhasil diubah".
Hasil uji	Mencetak pesan "Order berhasil diubah".
Status	Valid.

2. Kasus uji gagal mengubah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.75 Kasus uji gagal mengubah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah buat akta fidusia dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form ubah buat akta dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.45 Pengujian Validasi Membarui Status Buat Akta Fidusia

1. Kasus uji berhasil membarui status buat akta fidusia.

Tabel 6.76 Kasus uji berhasil membarui status buat akta fidusia

Nama Kasus Uji	Kasus uji berhasil membarui status buat akta fidusia.
Prosedur uji	1. Memilih data yang statusnya akan diperbarui. 2. Membarui status buat akta fidusia.
Hasil yang diharapkan	Status buat akta fidusia berhasil diperbarui.
Hasil uji	Status buat akta fidusia berhasil diperbarui.
Status	Valid.

6.3.46 Pengujian Validasi Mengonfirmasi Status Buat Akta Fidusia Selesai

1. Kasus uji berhasil membarui status buat akta fidusia selesai.

Tabel 6.77 Kasus uji berhasil membarui status buat akta fidusia selesai

Nama Kasus Uji	Kasus uji berhasil membarui status buat akta fidusia selesai.
Prosedur uji	1. Memilih data yang statusnya akan dikonfirmasi selesai. 2. mengonfirmasi status buat akta fidusia selesai.
Hasil yang diharapkan	Status buat akta fidusia berhasil dikonfirmasi selesai.
Hasil uji	Status buat akta fidusia berhasil dikonfirmasi selesai.
Status	Valid.

6.3.47 Pengujian Validasi Melihat Pendaftaran SK Fidusia

1. Kasus uji berhasil melihat pendaftaran SK fidusia.

Tabel 6.78 Kasus uji berhasil melihat pendaftaran SK fidusia

Nama Kasus Uji	Kasus uji berhasil melihat pendaftaran SK fidusia.
Prosedur uji	1. Memilih menu fidusia. 2. Memilih sub-menu pendaftaran SK.
Hasil yang diharapkan	Mempresentasikan data pendaftaran SK fidusia.
Hasil uji	Mempresentasikan data pendaftaran SK fidusia.
Status	Valid.



6.3.48 Pengujian Validasi Mengunggah Berkas SK Fidusia

1. Kasus uji berhasil mengunggah berkas SK fidusia.

Tabel 6.79 Kasus uji berhasil mengunggah berkas SK fidusia

Nama Kasus Uji	Kasus uji berhasil mengunggah berkas SK fidusia.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu fidusia. 2. Memilih sub-menu pendaftaran SK. 3. Memilih data berkas SK. 4. Memilih tombol unggah berkas SK. 5. Memilih menu <i>choose file</i>. 6. Memilih berkas yang ingin diunggah. 7. Memilih tombol <i>open</i>. 8. Memilih tombol <i>upload</i>.
Hasil yang diharapkan	Berkas SK Fidusia berhasil diunggah.
Hasil uji	Berkas SK Fidusia berhasil diunggah.
Status	Valid.

2. Kasus uji gagal mengunggah berkas SK fidusia dengan *file* yang diunggah bukan ekstensi .pdf.

Tabel 6.80 Kasus uji berhasil mengunggah berkas SK fidusia dengan file yang diunggah bukan ekstensi .pdf

Nama Kasus Uji	Kasus uji gagal mengunggah berkas SK fidusia dengan <i>file</i> yang diunggah bukan ekstensi .pdf.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu fidusia. 2. Memilih sub-menu pendaftaran SK. 3. Memilih data berkas SK. 4. Memilih tombol unggah berkas SK. 5. Memilih menu <i>choose file</i>. 6. Memilih berkas yang ingin diunggah bukan dengan ekstensi <i>file</i> .pdf. 7. Memilih tombol <i>open</i>. 8. Memilih tombol <i>upload</i>.
Hasil yang diharapkan	Berkas SK Fidusia gagal diunggah.
Hasil uji	Berkas SK Fidusia gagal diunggah.
Status	Valid.

6.3.49 Pengujian Validasi Melihat Rincian Berkas Klien pada Pendaftaran SK Fidusia

1. Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK fidusia.

Tabel 6.81 Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK fidusia

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada pendaftaran SK fidusia.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu fidusia. 2. Memilih sub-menu pendaftaran SK. 3. Memilih klien. 4. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada pendaftaran SK fidusia.
Hasil uji	Mempresentasikan rincian berkas klien pada pendaftaran SK fidusia.
Status	Valid.

6.3.50 Pengujian Validasi Melihat Rincian Berkas Klien pada Protokol Notaris

1. Kasus uji berhasil melihat rincian berkas klien pada protokol notaris.

Tabel 6.82 Kasus uji berhasil melihat rincian berkas klien pada protokol notaris

Nama Kasus Uji	Kasus uji berhasil melihat rincian berkas klien pada protokol notaris.
Prosedur uji	<ol style="list-style-type: none"> 1. Memilih menu protokol notaris. 2. Memilih klien. 3. Memilih tombol rincian berkas klien.
Hasil yang diharapkan	Mempresentasikan rincian berkas klien pada protokol notaris.
Hasil uji	Mempresentasikan rincian berkas klien pada protokol notaris.
Status	Valid.

6.3.51 Pengujian Validasi Mengubah Protokol Notaris

1. Kasus uji berhasil mengubah protokol notaris.

Tabel 6.83 Kasus uji berhasil mengubah protokol notaris

Nama Kasus Uji	Kasus uji berhasil mengubah protokol notaris.
Prosedur uji	<ol style="list-style-type: none"> 1. Mengubah data protokol notaris. 2. Memilih tombol ubah.
Hasil yang diharapkan	Data protokol notaris berhasil diubah.
Hasil uji	Data protokol notaris berhasil diubah.

Tabel 6.83 Kasus uji berhasil mengubah protokol notaris (lanjutan)

Nama Kasus Uji	Kasus uji berhasil mengubah protokol notaris.
Status	Valid.

2. Kasus uji gagal mengubah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.84 Kasus uji gagal mengubah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah protokol notaris protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	<ol style="list-style-type: none"> 1. Tidak mengisi form ubah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.52 Pengujian Validasi Menambah Protokol Notaris

1. Kasus uji berhasil menambah protokol notaris.

Tabel 6.85 Kasus uji berhasil menambah protokol notaris

Nama Kasus Uji	Kasus uji berhasil menambah protokol notaris.
Prosedur uji	<ol style="list-style-type: none"> 1. Mengisi form tambah protokol notaris. 2. Memilih tombol selesai.
Hasil yang diharapkan	Mencetak pesan "Protokol notaris berhasil ditambah".
Hasil uji	Mencetak pesan "Protokol notaris berhasil ditambah".
Status	Valid.

2. Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.86 Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.
-----------------------	--

Tabel 6.86 Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem (lanjutan)

Nama Kasus Uji	Kasus uji gagal menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form menambah protokol notaris dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol selesai.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

3. Kasus uji gagal menambah akun dengan nomor protokol yang sudah terdapat pada daftar protokol notaris.

Tabel 6.87 Kasus uji gagal menambah protokol notaris dengan nomor protokol yang sudah terdapat pada daftar protokol notaris

Nama Kasus Uji	Kasus uji gagal menambah protokol notaris dengan nomor protokol yang sudah terdapat pada daftar protokol notaris.
Prosedur uji	1. Mengisi form tambah protokol notaris. 2. Mengisi kolom nomor protokol yang sudah terdapat pada daftar protokol notaris. 3. Memilih tombol selesai.
Hasil yang diharapkan	Mencetak pesan "Nomor protokol telah digunakan".
Hasil uji	Mencetak pesan "Nomor protokol telah digunakan".
Status	Valid.

6.3.53 Pengujian Validasi Melihat Protokol Notaris

1. Kasus uji berhasil melihat protokol notaris.

Tabel 6.88 Kasus uji berhasil melihat protokol notaris

Nama Kasus Uji	Kasus uji berhasil melihat protokol notaris.
Prosedur uji	1. Memilih menu protokol notaris.
Hasil yang diharapkan	Mempresentasikan data protokol notaris.
Hasil uji	Mempresentasikan data protokol notaris.
Status	Valid.

6.3.54 Pengujian Validasi Melihat Dashboard

1. Kasus uji berhasil melihat *dashboard*.

Tabel 6.89 Kasus uji berhasil melihat *dashboard*

Nama Kasus Uji	Kasus uji berhasil melihat <i>dashboard</i> .
Prosedur uji	1. Aktor telah berhasil login sebagai pemilik atau pegawai.
Hasil yang diharapkan	Mempresentasikan halaman <i>dashboard</i> .
Hasil uji	Mempresentasikan halaman <i>dashboard</i> .
Status	Valid.

6.3.55 Pengujian Validasi Melihat Monitoring Kerja

1. Kasus uji berhasil melihat monitoring kerja.

Tabel 6.90 Kasus uji berhasil melihat monitoring kerja

Nama Kasus Uji	Kasus uji berhasil melihat monitoring kerja.
Prosedur uji	1. Memilih menu monitoring kerja.
Hasil yang diharapkan	Mempresentasikan data monitoring kerja.
Hasil uji	Mempresentasikan data monitoring kerja.
Status	Valid.

6.3.56 Pengujian Validasi Melihat Log Aktivitas Pegawai

1. Kasus uji berhasil melihat log aktivitas pegawai.

Tabel 6.91 Kasus uji berhasil melihat log aktivitas pegawai

Nama Kasus Uji	Kasus uji berhasil melihat log aktivitas pegawai.
Prosedur uji	1. Memilih menu log aktivitas pegawai.
Hasil yang diharapkan	Mempresentasikan data log aktivitas pegawai.
Hasil uji	Mempresentasikan data log aktivitas pegawai.
Status	Valid.

6.3.57 Pengujian Validasi Melihat Notifikasi *Deadline*

1. Kasus uji berhasil melihat notifikasi *deadline*.

Tabel 6.92 Kasus uji berhasil melihat notifikasi *deadline*

Nama Kasus Uji	Kasus uji berhasil melihat notifikasi <i>deadline</i> .
-----------------------	---

Tabel 6.92 Kasus uji berhasil melihat notifikasi *deadline* (lanjutan)

Nama Kasus Uji	Kasus uji berhasil melihat notifikasi <i>deadline</i> .
Prosedur uji	1. Aktor memilih tombol notifikasi <i>deadline</i> .
Hasil yang diharapkan	Mempresentasikan notifikasi <i>deadline</i> order.
Hasil uji	Mempresentasikan notifikasi <i>deadline</i> order.
Status	Valid.

2. Kasus uji berhasil melihat notifikasi *deadline* dengan tidak ada notifikasi *deadline*.

Tabel 6.93 Kasus uji berhasil melihat notifikasi *deadline* dengan tidak ada notifikasi *deadline*

Nama Kasus Uji	Kasus uji berhasil melihat notifikasi umur kertas dengan tidak ada notifikasi <i>deadline</i> .
Prosedur uji	1. Aktor memilih tombol notifikasi <i>deadline</i> .
Hasil yang diharapkan	Mencetak notifikasi notifikasi <i>deadline</i> "Tidak ada info <i>deadline</i> ".
Hasil uji	Mencetak notifikasi notifikasi <i>deadline</i> "Tidak ada <i>deadline</i> ".
Status	Valid.

6.3.58 Pengujian Validasi Melihat Notifikasi Umur Kertas

1. Kasus uji berhasil melihat notifikasi umur kertas.

Tabel 6.94 Kasus uji berhasil melihat notifikasi umur kertas

Nama Kasus Uji	Kasus uji berhasil melihat notifikasi umur kertas.
Prosedur uji	1. Aktor memilih tombol notifikasi umur kertas.
Hasil yang diharapkan	Mempresentasikan notifikasi notifikasi umur kertas protokol notaris.
Hasil uji	Mempresentasikan notifikasi notifikasi umur kertas protokol notaris.
Status	Valid.

2. Kasus uji berhasil melihat notifikasi umur kertas dengan tidak ada notifikasi umur kertas.

Tabel 6.95 Kasus uji berhasil melihat notifikasi umur kertas dengan tidak ada notifikasi umur kertas

Nama Kasus Uji	Kasus uji berhasil melihat notifikasi umur kertas dengan tidak ada notifikasi umur kertas
Prosedur uji	1. Aktor memilih tombol notifikasi umur kertas.
Hasil yang diharapkan	Mencetak notifikasi notifikasi umur kertas “Tidak ada info kerusakan”.
Hasil uji	Mencetak notifikasi notifikasi umur kertas “Tidak ada info kerusakan”.
Status	Valid.

6.3.59 Pengujian Validasi Menambah Informasi

1. Kasus uji berhasil menambah informasi.

Tabel 6.96 Kasus uji berhasil menambah informasi

Nama Kasus Uji	Kasus uji berhasil menambah informasi.
Prosedur uji	1. Mengisi form tambah informasi. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Informasi berhasil ditambah”.
Hasil uji	Mencetak pesan “Informasi berhasil ditambah”.
Status	Valid.

2. Kasus uji gagal menambah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.97 Kasus uji gagal menambah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal menambah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form menambah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol tambah.
Hasil yang diharapkan	Mencetak pesan “Please fill out this field”.
Hasil uji	Mencetak pesan “Please fill out this field”.
Status	Valid.

6.3.60 Pengujian Validasi Mengubah Informasi

1. Kasus uji berhasil mengubah informasi.

Tabel 6.98 Kasus uji berhasil mengubah informasi

Nama Kasus Uji	Kasus uji berhasil mengubah informasi.
Prosedur uji	1. Mengubah data informasi. 2. Memilih tombol ubah.
Hasil yang diharapkan	Data informasi berhasil diubah.
Hasil uji	Data informasi berhasil diubah.
Status	Valid.

2. Kasus uji gagal mengubah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.

Tabel 6.99 Kasus uji gagal mengubah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem

Nama Kasus Uji	Kasus uji gagal mengubah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem.
Prosedur uji	1. Tidak mengisi form mengubah informasi dengan terdapat kekosongan pada data yang dibutuhkan sistem. 2. Memilih tombol ubah.
Hasil yang diharapkan	Mencetak pesan "Please fill out this field".
Hasil uji	Mencetak pesan "Please fill out this field".
Status	Valid.

6.3.61 Pengujian Validasi Menghapus Informasi

1. Kasus uji berhasil menghapus informasi.

Tabel 6.100 Kasus uji berhasil menghapus informasi

Nama Kasus Uji	Kasus uji berhasil menghapus informasi.
Prosedur uji	1. Memilih menu informasi. 2. Memilih tombol hapus pada informasi yang dipilih. 3. Memilih tombol hapus. 4. Memilih tombol "Ya, Hapus!"
Hasil yang diharapkan	Informasi berhasil diubah.
Hasil uji	Informasi berhasil diubah.
Status	Valid.

2. Kasus uji gagal menghapus informasi.

Tabel 6.101 Kasus uji gagal menghapus informasi

Nama Kasus Uji	Kasus uji gagal menghapus informasi.
Prosedur uji	1. Aktor Memilih menu informasi. 2. Memilih tombol hapus pada informasi yang dipilih. 3. Memilih tombol hapus. 4. Memilih tombol "Cancel"
Hasil yang diharapkan	Informasi yang dipilih tidak dihapus.
Hasil uji	Informasi yang dipilih tidak dihapus.
Status	Valid.

6.4 Pengujian *Usability*

Pengujian *usability* dilakukan selepas proses tahap implementasi selesai. Pengujian *usability* digunakan cara uji coba sistem langsung terhadap responden agar mengetahui kualitas kemudahan penggunaan terhadap suatu sistem. *System Usability Scale (SUS)* merupakan metode yang digunakan untuk pengujian *usability*. *SUS* adalah suatu pengujian *usability* yang efektif dan andal apabila diterapkan terhadap bermacam produk dan aplikasi (Bangor, Kortum, & Miller, 2009). Pertanyaan yang digunakan ialah pertanyaan yang telah mengacu terhadap penelitian sebelumnya yang dilakukan oleh Said Atharillah Alifka Alhabsyi, Fajar Pradana, Fitra A. Bachtiar dengan judul "Pengembangan Sistem Point of Sale Berbasis Web Pada Edds Waffle Untuk Mendukung Penjualan Restaurant" Alhabsyi, et al. 2019).

6.4.1 Prosedur Pengujian *Usability*

Pertanyaan kuesioner terhadap pengguna merupakan prosedur yang dijalankan pada pengujian *usability*. Pengguna pada pengujian *usability* adalah pihak dari Kantor Notaris Dewi Andriani, S.H., M.H. Sebelum menjawab tiap pertanyaan, pengguna akan mencoba sistem terlebih dahulu. Sistem dicoba oleh lima pengguna yang terdiri atas satu pemilik dan empat pegawai kantor notaris. Setelah proses mencoba sistem selesai, kelima pengguna akan menjawab pertanyaan kuesioner yang sudah dipersiapkan. Pada Tabel 6.102 merupakan daftar pertanyaan *SUS*.

Tabel 6.102 Daftar Pertanyaan *SUS*

No.	Pertanyaan
1	Saya dapat mengoperasikan sistem ini dengan mudah.
2	Cara melakukan aksi pada sistem cukup sulit.
3	Informasi yang diberikan dibutuhkan oleh kantor notaris.

Tabel 6.102 Daftar Pertanyaan SUS (lanjutan)

No.	Pertanyaan
4	Saya merasa perlu pendampingan dari orang yang paham untuk dapat menggunakan sistem ini
5	Bahasa dan instruksi yang diberikan mudah dipahami
6	Saya merasa huruf pada aplikasi terlalu kecil
7	Saya suka dengan warna yang digunakan
8	Saya merasa tombol terlalu berdekatan sehingga akan sering salah menekan tombol
9	Jika saya akan menjalankan bisnis kantor saya, saya cenderung akan menggunakan sistem ini
10	Saya tidak akan menggunakan sistem ini

6.4.2 Analisis dan Hasil Pengujian *Usability*

Apabila prosedur pengujian *usability* telah dijalankan, selanjutnya akan dijalankan analisis terhadap hasil responden yang telah diperoleh. Jawaban pada setiap pertanyaan yang tersedia ditunjukkan pada Tabel 6.103.

Tabel 6.103 Hasil kuesioner skor tiap pertanyaan SUS

Nama	Dewi Andriani	Refni Nailiusna	Annin Dessy Anjani	Eka Pratiwi Rusri	Farhan Abdullah
No					
1	5	4	5	4	4
2	2	2	2	2	1
3	5	5	5	5	5
4	2	2	1	2	2
5	5	4	5	5	4
6	2	2	2	2	2
7	5	4	5	4	4
8	3	3	2	3	3
9	5	5	5	5	5
10	1	1	1	1	1

Setelah hasil skala *likert* didapatkan dari responden, hasil akan diolah melalui prosedur konversi skor agar memperoleh hasil akhir dari pengujian *usability* memanfaatkan metode SUS. Penilaian akan dilakukan pengurangan satu nilai dari respon yang didapat dari responden untuk pertanyaan dengan nomor 1, 3, 5, 7, 9, 10 dan dilakukan pengurangan lima untuk pertanyaan dengan nomor 2, 4, 6, 8, 10.

Setelah itu akan dilakukan penghitungan skor akhir dengan dikalikan 2.5, sehingga didapatkan skor SUS. Pada skor SUS dianalisis menerapkan penerimaan. Hasil dari konversi skor setiap pertanyaan ditunjukkan pada Tabel 6.104.

Tabel 6.104 Hasil kuesioner skor tiap pertanyaan SUS

Nama	Dewi Andriani	Refni Nailiusna	Annin Dessy Anjani	Eka Pratiwi Rusri	Farhan Abdullah	
No						
1	4	3	4	3	3	
2	3	3	3	3	4	
3	4	4	4	4	4	
4	3	3	4	3	3	
5	4	3	4	4	3	
6	3	3	3	3	3	
7	4	3	4	3	3	
8	2	2	3	2	2	
9	4	4	4	4	4	
10	4	4	4	4	4	
Jumlah	35	32	38	33	33	Rata-Rata
Dikali 2.5	87,5	80	95	82,5	82,5	85,5

Berdasarkan dari hasil yang telah diperoleh dari lima responden skor *likert* yang didapat, lalu dijalankan analisis dan hasil dari pengujian. *Usability* yang dihitung memerlukan pengujian *post study*, yaitu dengan metode *System Usability Scale* dengan skor akhir yang telah didapat ialah 85,5. Dari hasil skor akhir yang telah didapat sistem dapat dikelompokkan ke dalam kelompok *acceptable* yang berarti sistem telah sesuai dengan kebutuhan Kantor Notaris Dewi Andriani, S.H., M.H.

BAB 7 PENUTUP

7.1 Kesimpulan

Berasaskan terhadap hasil yang sudah didapatkan dari tahap analisis kebutuhan, perancangan, implementasi, dan pengujian sistem dapat dirumuskan bahwa:

1. Berasaskan hasil pada tahap analisis kebutuhan sistem pembukuan dan administrasi protokol notaris, enam puluh dua kebutuhan fungsional dan satu kebutuhan non-fungsional menjadi hasil yang diperoleh pada tahap ini. Selain itu, pengguna, pemilik, dan pegawai ialah aktor yang berpartisipasi pada sistem. Hasil analisis dari studi literatur, wawancara, dan observasi telah disesuaikan dengan kebutuhan yang dibutuhkan oleh kantor notaris.
2. Berasaskan hasil analisis kebutuhan menyusun hasil tahap perancangan. Perancangan *sequence diagram*, perancangan *class diagram*, perancangan algoritme, perancangan basis data, dan perancangan antarmuka merupakan hasil yang diperoleh dari tahap perancangan.
3. Berasaskan pada tahap implementasi didapatkan hasil sistem pembukuan dan administrasi protokol notaris menghasilkan spesifikasi pengembangan terhadap sistem, implementasi basis data, implementasi kode program, dan implementasi antarmuka.
4. Berasaskan pada tahap pengujian yang telah dijalankan, pengujian pada sistem menerapkan pengujian unit, pengujian integrasi, dan pengujian validasi. Pengujian unit dan integrasi menggunakan metode *whitebox testing* sedangkan untuk pengujian validasi menggunakan metode *blackbox testing*. Pengujian unit dijalankan terhadap tiga fungsi. Sementara itu, pengujian validasi dilakukan terhadap 93 kasus uji fungsional dan didapatkan hasil 100% valid. Pengujian non-fungsional, yaitu pengujian *usability* mendapatkan skor 85,5 dan dari skor ini dapat dikategorikan *acceptable*, sehingga sistem telah memenuhi kebutuhan pengguna kantor notaris.

7.2 Saran

Beberapa saran yang bisa dimungkinkan dalam pengembangan sistem pembukuan dan administrasi protokol notaris antara lain:

1. Menambahkan kebutuhan agar dapat membuat order notaris terhadap bidang PPAT.
2. Menambahkan kebutuhan untuk dapat mengelola audit terhadap kantor notaris.
3. Menambahkan aktor baru, yaitu klien untuk dapat konsultasi *online* dengan pemilik notaris.

DAFTAR REFERENSI

- Bangor, A., Kortum, P., & Miller, J. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*.
- Brooke John. 2011. Measuring Usability With The System Usability Scale (SUS). <http://www.measuringu.com/sus.php>. Diakses pada tanggal 22 Desember 2017.
- Dharwiyanti, S. & Wahono, R. S., 2003. Pengantar Unified Modeling Language (UML).
- Edmon, 2014. Notaris Bingung Dimana Harus Menyimpan Protokol Notaris", <http://www.hukumonline.com/berita/baca/lt532c49f3cbb01/n-otaris-bingung-dimana-harus-menyimpan-protokol-notaris>. Diakses pada 14 Agustus 2018 pukul 21.04.
- Hakim, L., 2010. Membangun Web Berbasis PHP dengan Framework Codeigniter. Dalam: Yogyakarta: Lokomedia.
- Hendini, A., 2016. Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak).
- Kuswanto, Mohamat Riza. 2017. Urgensi Penyimpanan Protokol Notaris Dalam Bentuk Elektronik dan Kepastian Hukumnya di Indonesia, Surakarta: Universitas Sebelas Maret.
- Lightowler, Michael. 2007. E-Commerce For Notaries In England & Wales. England.
- Pemerintah Indonesia. 2014. *Undang-undang Nomor 2 Tahun 2014 tentang Perubahan atas Undang-Undang Nomor 30 Tahun 2014 tentang Jabatan Notaris*. Lembaran RI Tahun 2014 No.3 Jakarta: Sekretariat Negara.
- Pressman, R. S., 2010. *Software Engineering*. 7th ed. New York: McGraw-Hill Companies, Inc.
- Pressman R. S., 2010. *Software Engineering*. 7th ed. New York: McGraw-Hill Companies, Inc.
- Pressman R. S., 2010. *Software Engineering*. 5th ed. New York: McGraw-Hill Companies, Inc.
- Riyandwyana, A. & Mukhlason, E. S. A., 2012. Pengembangan Sistem Rekomendasi Rekomendasi Peminjaman Buku Berbasis Web Menggunakan Metode Self Organizing Map Clustering Pada Badan Perpustakaan Dan Kearsipan (BAPERSIP) Provinsi Jawa Timur. *Jurnal Teknik ITS*, pp. A-376.
- Rosa, A. & Shalahuddin, M., 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.
- Shah, M., 2014. *Impact of Management Information Systems (MIS) on School Administration: What the Literature Says*. *Procedia - Social and Behavioral Sciences*, [daring] 116 hal.2799–2804. Tersedia di:

<http://linkinghub.elsevier.com/retrieve/pii/S1877042814006764> [Diakses 15 Agustus 2017].

Sharma, K. S. D. & Singh, A., 2016. Software Development Life Cycle. International Journal of Modern Engineering and Research Technology, p. 5.

Siagian. 2001:267. "Administrasi Perkantoran". Jakarta: Penerbit Gunung Agung.

Sommerville, I., 2011. Software Engineering. 9th ed. USA: Pearson Education, Inc.

Spillner, Andreas, Tilo Linz, and Hans Schaefer. 2014. Software Testing Foundations - A Study Guide for the Certified Tester Exam. Edited by Dr. Michael Barabas. Heidelberg, Germany. Vol. 16. Santa Barbara, CA.



LAMPIRAN A SURAT BUKTI ELISITASI KEBUTUHAN



NOTARIS & PPAT
DEWI ANDRIANI, SH.MH.
Kantor : Jl. Balai Pustaka Timur B2 - No. 43
Rawamangun - Jakarta Timur
Telp .(021) 47865077 Fax. (021) 47868056
e-mail : notarisdewi2000@yahoo.com
JAKARTA - 13220

Jakarta, 8 April 2019

No : 03 / N / IV / 2019
Sifat : Biasa
Lampiran : -
Hal : **Surat Keterangan Persetujuan Penelitian**

Kepada Yth.
Jurusan Teknik Informatika
Universitas Brawijaya
Di tempat

Dengan Hormat,

Yang bertanda tangan di bawah ini, menerangkan bahwa:

Nama : Krishna Yudhakusuma Putra Munandar
Nim : 155150207111004
Jurusan/Prodi : S1 Teknik Informatika
Konsentrasi : Rekayasa Perangkat Lunak

Adalah benar telah di terima dan melakukan Penelitian di Kantor Notaris – PPAT Dewi Andriani, SH., MH dengan judul **“Pengembangan Sistem Pembukuan dan Administrasi Protokol Notaris”** dengan Periode 16 Juli 2018 – 19 Desember 2018. Adapun penelitian yang dilakukan meliputi:

1. Elisitasi Kebutuhan
2. Tes Usabilitas

Demikian surat keterangan ini dibuat agar dapat digunakan sebagaimana mestinya. Atas perhatian dan kerjasamanya diucapkan terima kasih.

Notaris, PPAT


DEWI ANDRIANI, SH.MH.

LAMPIRAN B HASIL WAWANCARA

Narasumber : Dewi Andriani, S.H., M.H.

Posisi : Pemilik Kantor Notaris

1. Bagaimana proses bisnis ketika dari awal klien pertama kali datang sampai dengan menyelesaikan aktanya.

Bahwa ketika klien datang biasanya konsultasi utk memberi tahu apa yang klien inginkan. Nanti setelah itu notaris akan memberika syarat syarat dan kelengkapan dan data data apa yang diperlukan. Kemudian setelah itu biasanya klien akan datang lagi membawa persyaratan yang dibutuhkan. Contohnya misalnya klien itu akan membuat akta kuasa menjual, maka yang diperlukan adalah fotokopi ktp, kartu keluarga, surat nikah, pemberi kuasa, fotokopi sertifikat, dan pbb serta fotokopi ktp dan kartu keluarga penerima kuasa. Nanti selanjutnya notaris akan membuatkan minuta aktanya dan setelah itu kita akan jadwalkan penandatanganan. Selanjutnya pada hari yang sudah ditentukan klien akan datang ke kantor lalu akan ditandatangani oleh para pihak. Disinilah mulai proses orderan tersebut atau pekerjaan tersebut dimulai dari kantor saya.

Yang pertama kali dikerjakan oleh staff saya dimana akan dibuatkan salinan aktanya, setelah di print di tandatangani, dijahit, dilampiri kelengkapan dokumennya. Kemudian disimpan sebagai protokol notaris, sedangkan salinan aktanya akan diberikan kepada klien.

2. Pewawancara : Bagaimana proses bisnis dalam mengelola akta yang disimpan?

Dalam pengelolaan akta yang disimpan, biasanya akan kita kumpulkan dulu semua akta-akta tersebut sampai satu bulan, setelah itu akan dijahit dan dibundel dalam bentuk bundel akta perbulannya. Lalu disimpan dalam brankas lemari besi yang tahan air dan api yang tujuannya adalah apabila terjadi segala sesuatu maka berkas tersebut tidak akan rusak. Permasalahannya yang muncul adalah apabila akta yang kita buat ini banyak misalnya satu tahun itu sampai 200 atau 400 sehingga memerlukan lemari brankas dan ruangan yang cukup besar penyimpanannya serta memerlukan lemari dan lainnya sehingga membutuhkan dana-dana lain yang maksudnya ekstra yang harus kita persiapkan.

3. Pewawancara : Bagaimana proses bisnis untuk mengelola data klien?

Untuk selama ini dalam mengelola data klien kami masih manual, artinya mencatat secara manual pada buku register yang biasanya disebut repositorium akta. Kekurangan dari sistim ini adalah apabila klien kehilangan akta yang pernah dibuat misalnya 10 tahun yang lalu hilang lalu datang lagi untuk minta salinan dan mereka tidak mengetahui lagi tanggal berapa, bulan, serta nomor akta yang pernah dibuat di kantor saya maka kita pun akan sulit untuk mencarinya karena kami harus membuka lemari satu-satu, memperkirakan kapan akta tersebut mereka buat dan itu memakan waktu serta

berkemungkinan kecil untuk mendapatkannya akta yang dicari itu dapat dengan cepat seperti itu.

4. Pewawancara : Bagaimana proses bisnis untuk mengelola data pegawai?

Kalau untuk pengelolaan data pegawai tidak terlalu sulit, kami cukup memiliki cv saat melamar pekerjaan dan dimasukkan ke bundel kerja. Karena karyawan kan tidak terlalu banyak seperti jumlah akta.

5. Pewawancara : Bagaimana proses bisnis untuk mengelola protokol notaris?

Untuk pengelolaan protokol notaris sesuai dengan aturan undang-undang jabatan notaris dimana diharuskan dijahit dan dibundel setiap bulannya lalu disimpan di lemari besi disuatu tempat. Masalah yang timbul adalah apabila notaris itu seperti saya sudah bekerja kurang lebih 20 tahun dengan akta setiap bulannya kurang lebih 50. Bisa dibayangkan bagaimana menyimpan protokol tersebut apalagi bila protokol tersebut saya simpan masa 2ension masih 20 tahun lagi. Itu agak repot dan memerlukan tempat dan dana yang tidak sedikit untuk biaya penyimpanan dan memerlukan perawatan yang ekstra agar tidak rusak dimakan rayap dan warna kertas akta tidak kuning. Seperti itu.

6. Pewawancara : Apakah selama ini proses bisnis yang ada ini efektif?

Menurut saya proses manual saat ini sudah tidak efektif ya dengan kemajuan jaman sekarang karena dengan semakin lama notaris praktek tentu makin banyak protokol yang ia simpan dan memerlukan metode tersistim yang bisa memperoleh pencarian protokol notaris yang sistemnya secara online atau memerlukan suatu metode tersistim yang bisa kita lihat untuk seluruh oleh notaris dan karyawan dengan memakai apaya.... metode online... gitu ya, jadi data disimpan didalam suatu database dimana tinggal kita klik kita akan menemukan apa yang klien perlukan tersebut.

7. Berapa banyak akta yang dapat diselesaikan dalam waktu seminggu?

Biasanya akta yang kita selesaikan dalam seminggu itu bisa sampai 4, 5, atau 6 ya tergantung akta yang kita buat.

8. Apa saja kendala di kantor notaris?

Kendala yang biasanya ditemui umumnya di notaris dengan sistem manual ini adalah sulit untuk mengontrol proses pekerjaan yang sudah sampai mana, jadi kita tidak mengetahui baik notaris maupun staff. Klien atas nama a atau b itu sudah sampai mana prosesnya, kecuali kita melihat berkasnya atau data-datanya. Kemudian pekerjaan tidak selesai sesuai jadwal karena terkadang staff suka lupa mana yang harus didahulukan sehingga lewat dari waktu yang diperjanjikan. Kemudian sulit juga kadang kadang mencari berkas dan minuta akta apabila secara manual karena bisa saja protokol tersebut tercecer, hilang, rusak atau warna kertasnya sudah kuning atau bisa kejadian seperti kena musibah banjir, gempa bumi sementara notaris, kami tidak memiliki database nama nama klien, serta tidak punya data data akta yang sudah kami buat selama ini.

9. Sistem atau aplikasi apa yang diharapkan dari kantor notaris?

Sistim yang mungkin kami perlukan adalah sistem monitoring kerja pekerjaan yang masuk, lalu pengingat tenggat untuk pekerjaan, dan database nama nama klien yang sudah membuat akta berpuluh-puluh tahun yang lalu di kami namun masih busa ditemukan datanya saat ini sehingga tidak perlu lagi dibutuhkan penyimpanan pencarian untuk protokol notaris sehingga agar mudah untuk mencarinya dan dapat membackup data protokol tersebut. jadi juga dengan sistimnya ini ada monitoring dan database kemudian kami juga memerlukan dengan sistim membuka abjad misalnya abu, kita klik abu dan tanggal lahirnya dapatlah data data yang semua tentang beliau pernah buat di kantor kami, seperti itu yang mungkin kami perlukan untuk kedepannya.

Jakarta, 20 Desember 2018

Pemilik Kantor Notaris Dewi Andriani



Dewi Andriani, S.H., M.H.



LAMPIRAN C KUESIONER SYSTEM USABILITY SCALE

System Usability Scale

1. Pertanyaan dengan nomor 1, 3, 5, 7, 9 adalah pertanyaan hal positif.
2. Pertanyaan dengan nomor: 2, 4, 6, 8, 10 adalah pertanyaan hal negatif.

Silahkan pilih salah satu opsi dengan nilai antara 1 dan 5.

Keterangan nilai:
Nilai 1 = Sangat Tidak Setuju
Nilai 2 = Tidak Setuju
Nilai 3 = Netral
Nilai 4 = Setuju
Nilai 5 = Sangat Setuju

* Required

Nama Lengkap *

Your answer

Umur *

Your answer

Saya dapat mengoperasikan sistem ini dengan mudah. *

1	2	3	4	5
<input type="radio"/>				

Cara melakukan aksi pada sistem cukup sulit.

1	2	3	4	5
<input type="radio"/>				

Informasi yang diberikan dibutuhkan oleh kantor notaris. *

1	2	3	4	5
<input type="radio"/>				

Saya merasa perlu pendampingan dari orang yang paham untuk dapat menggunakan sistem ini *

1	2	3	4	5
<input type="radio"/>				

Bahasa dan instruksi yang diberikan mudah dipahami *

1	2	3	4	5
<input type="radio"/>				

Saya merasa huruf pada aplikasi terlalu kecil *

1	2	3	4	5
<input type="radio"/>				

Saya suka dengan warna yang digunakan

1	2	3	4	5
<input type="radio"/>				

Saya merasa tombol terlalu berdekatan sehingga akan sering salah menekan tombol *

1	2	3	4	5
<input type="radio"/>				

Jika saya akan menjalankan bisnis kantor saya, saya cenderung akan menggunakan sistem ini

1	2	3	4	5
<input type="radio"/>				

Saya tidak akan menggunakan sistem ini

1	2	3	4	5
<input type="radio"/>				

SUBMIT



LAMPIRAN D RESPONDEN SYSTEM USABILITY SCALE

Nama Lengkap

5 responses

Refni Nailiusna
Dewi Andriani
Annin
Eka Pratiwi Rusri
Farhan abdulah

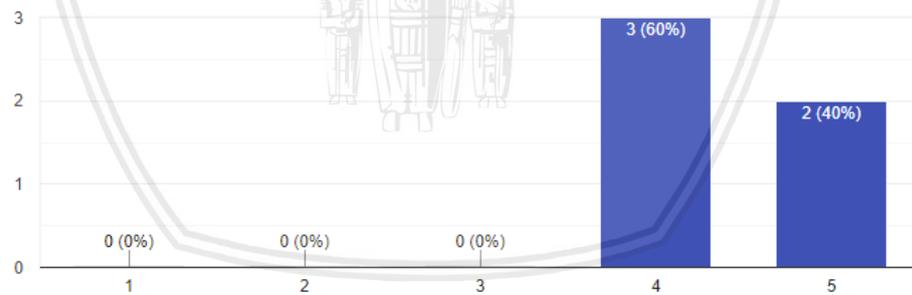
Umur

5 responses

34
48
25
29
21

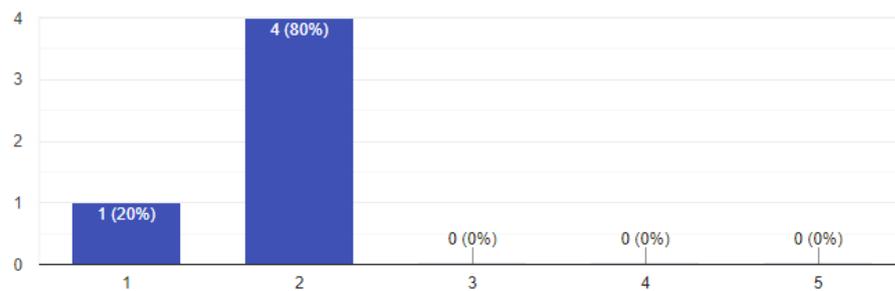
Saya dapat mengoperasikan sistem ini dengan mudah.

5 responses



Cara melakukan aksi pada sistem cukup sulit.

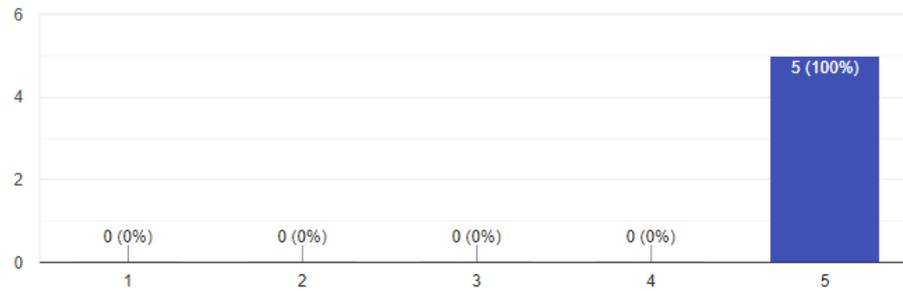
5 responses



Informasi yang diberikan dibutuhkan oleh kantor notaris.

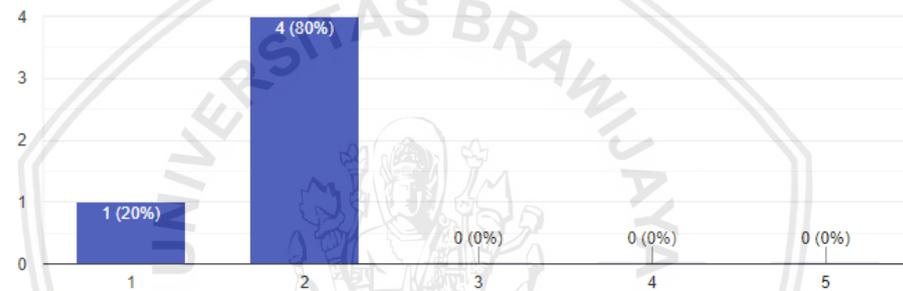


5 responses



Saya merasa perlu pendampingan dari orang yang paham untuk dapat menggunakan sistem ini

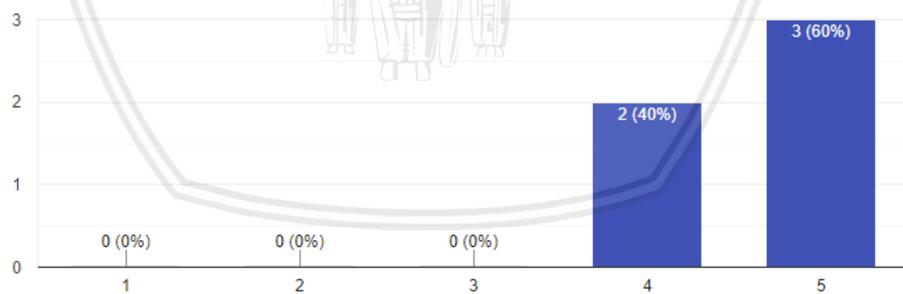
5 responses



Bahasa dan instruksi yang diberikan mudah dipahami

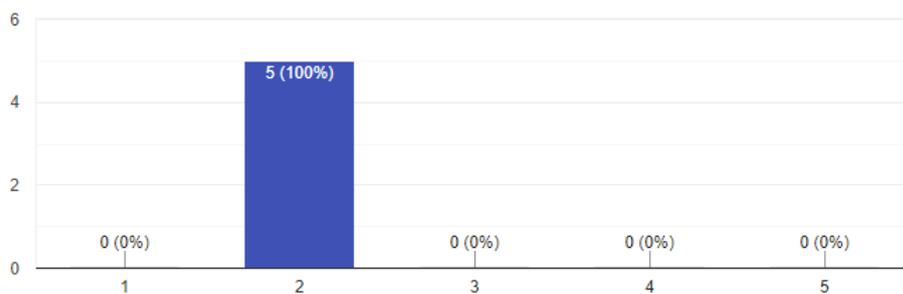


5 responses



Saya merasa huruf pada aplikasi terlalu kecil

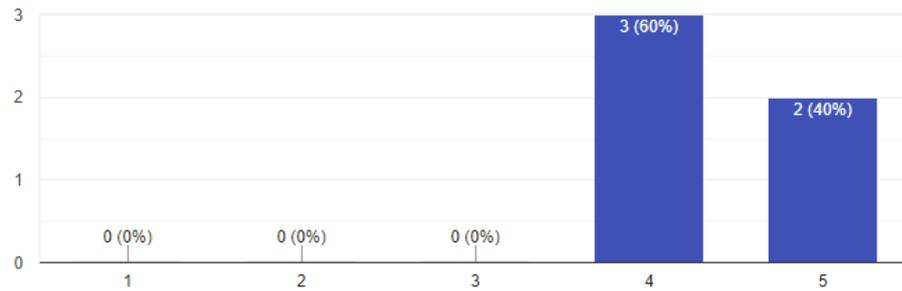
5 responses



Saya suka dengan warna yang digunakan

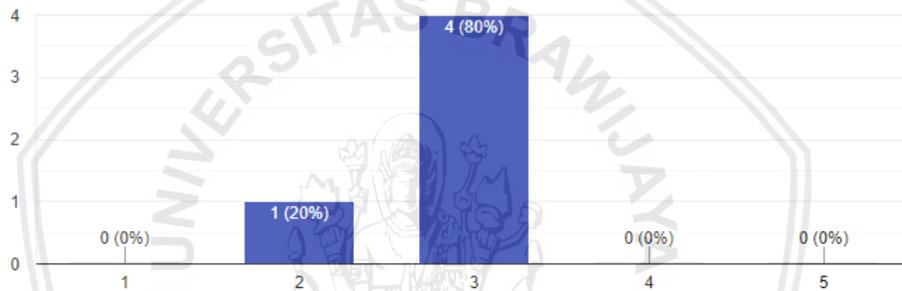


5 responses



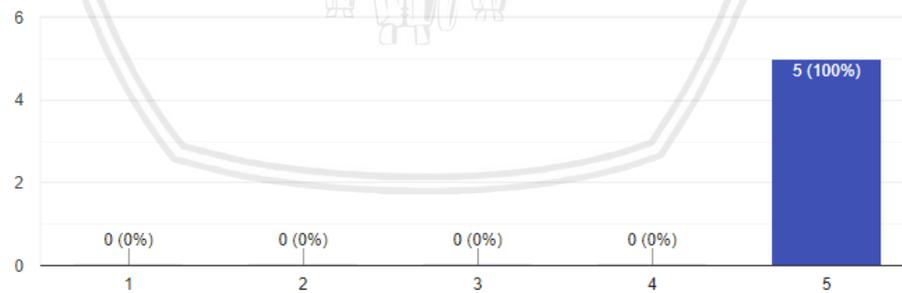
Saya merasa tombol terlalu berdekatan sehingga akan sering salah menekan tombol

5 responses



Jika saya akan menjalankan bisnis kantor saya, saya cenderung akan menggunakan sistem ini

5 responses



Saya tidak akan menggunakan sistem ini



5 responses

