

**PERANGKAT LUNAK PELAYANAN KLAIM ASURANSI KREDIT UNTUK
MEMPERMUDAH PENGAJUAN KLAIM DI DEPARTEMEN
PELAYANAN KLAIM PT. MULTI ARTHA
INSURANCE BROKERS**

LAPORAN PRAKTIK KERJA

Diajukan untuk memenuhi
salah satu persyaratan sidang tugas akhir

Disusun oleh

RIZKI PEBRIANTO

NRP. 160623001

KELAS. 3IFP-01



**POLITEKNIK KOMPUTER NIAGA LPKIA
PROGRAM STUDI MANAJEMEN INFORMATIKA
KONSENTRASI TEKNIK INFORMATIKA
BANDUNG
2018**

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini,

Nama : **Rizki Pebrianto**
N.R.P : 160623001
Program Studi : Manajemen Informatika Konsentrasi Teknik
Informatika STMIK LPKIA
Judul Laporan : *Perangkat Lunak Pelayanan Klaim Asuransi
Kredit Untuk Mempermudah Pengajuan Klaim Di
Departemen Pelayanan Klaim PT. Multi Artha
Insurance Brokers*

Dengan ini menyatakan bahwa hasil penulisan Laporan Praktik Kerja yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata di kemudian hari penulisan Laporan Praktik Kerja ini merupakan hasil plagiat atau penjiplakan terhadap karya orang lain, maka saya bersedia mempertanggungjawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di STMIK LPKIA.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Bandung, 18 Oktober 2018

Penulis,



Rizki Pebrianto

LEMBAR PERSETUJUAN

LAPORAN PRAKTIK KERJA

PERANGKAT LUNAK PELAYANAN KLAIM ASURANSI KREDIT UNTUK MEMPERMUDAH PENGAJUAN KLAIM DI DEPARTEMEN PELAYANAN KLAIM DI PT. MULTI ARTHA INSURANCE BROKERS

Diajukan untuk memenuhi
salah satu persyaratan sidang tugas akhir

Disusun oleh

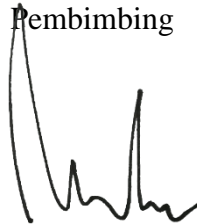
RIZKI PEBRIANTO
NRP.160623001

Telah diperiksa dan disetujui untuk mengikuti seminar program

Pada tanggal : 2 September 2018

di Bandung

Pembimbing



Andy Victor P, ST., MT

NIP. 106002

LEMBAR PENGESAHAN

LAPORAN PRAKTIK KERJA

PERANGKAT LUNAK PELAYANAN KLAIM ASURANSI KREDIT UNTUK MEMPERMUDAH PENGAJUAN KLAIM DI DEPARTEMEN PELAYANAN KLAIM PT. MULTI ARTHA INSURANCE BROKERS

Diajukan untuk memenuhi
salah satu persyaratan sidang tugas akhir

Disusun oleh

RIZKI PEBRIANTO
NRP.160623001

Telah diperiksa dan disetujui

Pada tanggal : 2 September 2018
di Bandung

Mengetahui

Pembimbing

Andy Victor P, ST.,MT

NIP. 106002



Ketua Program Studi

Andy Victor P, ST., MT

NIP. 106002

ABSTRAKSI

Rizki Pebrianto.160623001

**PERANGKAT LUNAK PELAYANAN KLAIM ASURANSI KREDIT UNTUK
MEMPERMUDAH PENGAJUAN KLAIM DI DEPARTEMEN PELAYANAN
KLAIM PT. MULTI ARTHA INSURANCE BROKERS**

Laporan Praktik Kerja. Program Studi Manajemen Informatika. Konsentrasi Teknik Informatika. 2018

Kata kunci : Pelayanan Klaim Asuransi

(27 lembar)

Laporan Praktek Kerja ini memuat semua perancangan yang penulis ajukan ke perusahaan tempat penulis melakukan Praktek Kerja Lapangan yaitu di PT. Multi Artha Insurance Brokers

Dibuatnya latar belakang laporan praktik kerja lapangan ini adalah membahas permasalahan pelayanan klaim asuransi yang dilakukan oleh bagian pelayanan klaim di PT. Multi Artha Insurance Broker dimana proses pengajuan klaim asuransi jiwa masih dilakukan secara tertulis dan menghabiskan banyak waktu seta dibantu oleh pihak lain seperti jasa pengiriman paket sebagai perantara pengiriman dokumen oleh bank untuk mengajukan asuransi jiwa kredit debiturnya, Maka penulis membantu dalam pembuatan suatu program pelayanan klaim asuransi berbasis web agar lebih efektif dan efisien.

Oleh karena adanya masalah ini penulis tertarik untuk melakukan perancangan sistem yang dapat mendukung pelaksanaan kerja karyawan pada divisi Pelayanan Klaim Asuransi di PT. Multi Artha Insurance Brokers

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat-Nya sehingga laporan praktik kerja ini dapat penulis selesaikan. Laporan praktik kerja ini merupakan salah satu syarat kelulusan program pendidikan Sarjana (S1) pada program studi Teknik Informatika STMIK LPKIA Bandung.

Laporan praktik kerja ini membuat suatu sistem yang berjudul “Perangkat Lunak Pelayanan Klaim Asuransi Kredit Untuk Mempermudah Pengajuan Klaim Di Departemen Pelayanan Klaim PT. Multi Artha Insurance Brokers” yang diambil berdasarkan hasil analisa saat praktik berlangsung . Penyusunan laporan praktik kerja ini tidak lepas dari dukungan bantuan dan bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan rahmat-Nya.
2. Orang tua yang saya hormati, serta seluruh keluarga yang telah membantu baik moril maupun materil selama ini.
3. Bapak Andy Victor MT., MOS., MCP., MTCNA. selaku Ketua Program Studi Manajemen Informatika STMIK LPKIA Bandung, sekaligus sebagai dosen pembimbing yang telah memberikan arahan dan masukan serta saran selama penyusunan laporan praktik kerja.
4. Teman-teman yang telah memberikan semangat dan memberikan solusi kepada penyusun.

Sebagai penutup penulis mengharapkan semoga laporan praktik kerja ini membawa manfaat khususnya bagi penulis dan umumnya bagi pihak yang membaca laporan ini, semoga Allah SWT selalu melimpahkan rahmat dan karunia-Nya kepada kita semua. Amiin

Bandung, Agustus 2018

Penulis,













Rizki Pebrianto

DAFTAR ISI





	Halaman
JUDUL HALAMAN	i
LEMBAR PERNYATAAN.....	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
ABSTRAKSI	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR SIMBOL	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah	1
1.2. Identifikasi Persoalan	2
1.3. Ruang Lingkup.....	3
1.4. Tujuan	3
1.5. Sistematika Penulisan	3
BAB II LANDASAN TEORI.....	5
2.1. Teori Tentang Permasalahan	5
2.1.1. Sistem.....	5
2.1.2. Informasi	8
2.1.3. Sistem Informasi	9
2.1.4. Asuransi.....	11
2.1.5. Klaim Asuransi.....	11
2.1.6. Data	11
2.1.7. <i>Unified Modelling Language (UML)</i>	12
2.1.8. Database	13
2.2. Bahasa Pemograman	14
2.2.1. Website	14
2.2.2. PHP (PHP Hypertext Preprocessor)	15
2.2.3. HTML (Hypertext Markup Language).....	16
2.2.4. <i>JavaScript</i>	17
2.2.5. <i>CodeIgniter</i>	19
2.3. Basis Data	20
2.3.1. MySQL	20
BAB III ANALISIS DAN PERANCANGAN	21
3.1. Gambaran Umum Perusahaan.....	21

4.1.1.	Profil Perusahaan	21
3.2.	Analisis Fungsional	23
3.2.1.	<i>Use Case</i> Diagram	24
3.2.2.	Deskripsi Dan Skenario <i>Use Case</i>	24
3.2.3.	Diagram Aktivitas (<i>Activiti Diagrams</i>).....	29
3.2.4.	Diagram Kelas (<i>Class Diagrams</i>).....	33
3.3.	Perancangan Data	33
3.4.	Perancangan Antarmuka	34
3.5.	Perancangan Prosedural	39
BAB IV IMPLEMENTASI DAN PENGUJIAN.....		41
4.1.	Implementasi Perangkat Lunak.....	41
4.1.1.	Lingkup dan Batasan	41
4.1.2.	Spesifikasi Perangkat Keras (<i>Hardware</i>)	41
4.1.3.	Spesifikasi Perangkat Lunak (<i>Software</i>)	42
4.1.4.	Hak Akses (<i>Brainware</i>)	42
4.1.5.	Implementasi Antarmuka	43
4.2.	Pengujian	46
4.2.1.	Pengujian <i>Blax Box</i> Pada Menu Login	46
4.2.2.	Pengujian <i>Blax Box</i> Pada Menu Data Debitur.....	48
BAB V KESIMPULAN DAN SARAN		50
5.1.	Kesimpulan.....	50
5.2.	Saran	50
DAFTAR PUSAKA.....		51

DAFTAR SIMBOL

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi






Gambar 1. 1 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>N-Ary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Gambar 1. 2 Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Gambar 1. 3 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Gambar 1. 4 Simbol Activity Diagram

DAFTAR GAMBAR

Gambar 1. 1 Simbol Use Case Diagram	ix
Gambar 1. 2 Simbol Class Diagram	x
Gambar 1. 3 Simbol Sequence Diagram	x
Gambar 1. 4 Simbol Activity Diagram	xi
Gambar 3. 1 Logo Perusahaan	22
Gambar 3. 2 Stuktur Organisasi	23
Gambar 3. 3 Use Case Diagram	24
Gambar 3. 4 Activity Diagram Mengelola Debitur	29
Gambar 3. 5 Activity Diagram Unggah Dokumen	30
Gambar 3. 6 Activity Diagram Mengajukan Klaim	30
Gambar 3. 7 Activity Diagram Unduh Dokumen peserta	31
Gambar 3. 8 Activity Diagram Validasi Pengajuan	31
Gambar 3. 9 Activity Diagram Mengirim Notifikasi	32
Gambar 3. 10 Activity Diagram Kelola Bank	32
Gambar 3. 11 Class Diagram	33
Gambar 3. 12 Perancangan data	33
Gambar 3. 13 Halaman Login	34
Gambar 3. 14 Halaman Dashboard	35
Gambar 3. 15 Halaman Dashboard Admin	35
Gambar 3. 16 Halaman Menu Debitur	36
Gambar 3. 17 Halaman Form Tambah Data Debitur	37
Gambar 3. 18 Halaman Pengajuan Klaim	37
Gambar 3. 19 Form Pengajuan Klaim	38
Gambar 3. 20 Halaman Produksi AJK	39

DAFTAR TABEL

Tabel 3. 1 Deskripsi Use Case Mengelola Debitur.....	25
Tabel 3. 2 Deskripsi Use Case Unggah Dokumen.....	26
Tabel 3. 3 Deskripsi Use Case Mengajukan Klaim	26
Tabel 3. 4 Deskripsi Use Case Unduh Dokumen	26
Tabel 3. 5 Deskripsi Use Case Memvalidasi Pengajuan Klaim.....	27
Tabel 3. 6 Deskripsi Use Case Mengelola Bank.....	28
Tabel 3. 7 Deskripsi Use Case Mengirimkan Notifikasi	28

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Broker asuransi merupakan jasa penghubung antara klien (tertanggung) dengan perusahaan asuransi (penanggung), dapat memberikan pilihan dan saran kepada klien yang masih kebingungan dalam memilih perusahaan asuransi mana yang tepat baginya dan juga mewakili klien dalam bernegosiasi dengan perusahaan asuransi dengan tujuan memperoleh cakupan pertanggung jawaban yang luas. Bukan itu saja, bila terjadinya pengajuan ganti rugi (klaim), klien tidak perlu repot mengurus sendiri ke perusahaan asuransi karena yang akan mengurus klaim tersebut adalah broker asuransi.

Pada asuransi kredit yang menjadi tertanggung adalah Bank/Lembaga Pembiayaan Keuangan yang mengajukan permintaan asuransi kredit bukan debitur yang meminjam dana dari Bank Umum/Lembaga Pembiayaan Keuangan tersebut. Dengan demikian asuransi kredit merupakan biparty agreement dimana hanya ada dua pihak yang terlibat yaitu perusahaan asuransi sebagai penanggung dan bank umum/lembaga pembiayaan sebagai tertanggung.

Asuransi kredit mempunyai kaitan erat dengan jasa perbankan terutama di bidang perkreditan yang selalu dikaitkan dengan jaminan kredit berupa barang bergerak dan tidak bergerak yang sewaktu-waktu dapat tertimpa risiko yang dapat mengakibatkan kerugian bagi pemilik barang dan bank sebagai pemberi kredit.

Kredit adalah pinjaman uang yang diberikan oleh pemberi kepada nasabahnya. Untuk melindungi diri dari kemungkinan nasabah yang tidak dapat mengembalikan kredit, pemberi kredit menutup asuransi atas kredit tersebut. Dalam asuransi kredit, yang menjadi pihak tertanggung adalah pemberi kredit (bank dan /atau lembaga keuangan) dan yang ditanggung oleh penanggung adalah risiko kredit di mana tidak diperolehnya kembali kredit kepada para nasabahnya.

Saat ini PT. Multi Artha Broker Insurance Brokers atau biasa disebut Maibro dalam melayani pengajuan klaim kliennya belum menggunakan Perangkat Lunak pelayanan klaim dalam bentuk program atau masih dibantu oleh perusahaan jasa pengiriman properti. Proses pengajuan klaim ini berawal dari klien / bank yang mengajukan klaim debiturnya dengan beberapa kelengkapan persyaratan dokumen klaim lalu dikirim melalui perusahaan jasa pengiriman lalu diterima oleh bagian pelayanan klaim PT. Maibro, terdapat beberapa persyaratan dokumen dalam mengajukan klaim dimana dari satu debitur memiliki banyak dokumen klaim yang menyebabkan menumpuknya dokumen di bagian pelayanan klaim Maibro sehingga menyulitkan dalam pengelolaan dokumen debitur yang diajukan.

Sehingga disinilah PT. Maibro membutuhkan yang namanya Perangkat Lunak yang berbasis teknologi komputer untuk memudahkan pengelolaan data atau dokumen serta pengajuan klaim tertanggung yang diajukan oleh bank/nasabah ke pihak perusahaan asuransi. Melihat permasalahan tersebut, maka penulis berusaha memberikan solusi untuk meningkatkan atau sedikitnya membantu untuk mempermudah pengelolaan pengajuan klaim asuransi yang diajukan oleh klien dengan cara membuat sebuah Perangkat Lunak pelayanan klaim asuransi jiwa kredit.

1.2. Identifikasi Persoalan

Adapun identifikasi persoalan dalam penanganan pelayanan klaim di PT.MULTI ARTHA INSURANCE BROKERS antara lain :

1. Belum adanya Perangkat Lunak pelayanan klaim dalam bentuk program.
2. Memakan waktu yang lama bagi klien yang mengajukan klaim asuransi debiturnya.
3. Sulitnya mengelola data dokumen klaim yang diajukan oleh klien baik yang sudah lengkap maupun belum lengkap.

1.3. Ruang Lingkup

Agar pembatasan masalah ini dapat dilakukan secara terarah dan sesuai dengan yang diharapkan, untuk itulah perlu membatasi ruang lingkup dari permasalahan ini. Ruang lingkup permasalahan diantaranya:

1. Aplikasi ini mengolah data nasabah / bertanggung di mulai dari data pribadi debitur.
2. Klaim yang diajukan meliputi data debitur dan dokumen serta program asuransi yang diikuti
3. Sebagai penyimpanan data klaim dan status klaimnya.
4. Aplikasi ini tidak memproses perhitungan premi, plafon, dan jumlah klaim yang harus dibayar oleh pihak asuransi.
5. Tidak adanya pembahasan tentang akseptasi dan rekapitulasi

1.4. Tujuan

Tujuan dari penelitian ini diantaranya yaitu :

1. Dibuatnya perangkat lunak untuk mempermudah klien / bank dalam mengajukan klaim asuransi kredit debiturnya
2. Mempercepat waktu pengajuan klaim sekaligus meningkatkan waktu untuk mengelola dan mencari data nasabah yang belum memenuhi persyaratan klaim
3. Mempermudah klient untuk melihat status pengajuan klaim asuransi kredit debiturnya

1.5. Sistematika Penulisan

Untuk memudahkan dalam mengikuti seluruh uraian dan pembahasan atas laporan praktik kerja ini maka penulisan laporan praktik kerja ini disusun dengan sistematika penulisan sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini membahas tentang Latar Belakang Masalah, Identifikasi Persoalan, Ruang Lingkup, Tujuan Pembuatan Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers.

2. BAB II LANDASAN TEORI

Bab ini membahas tentang teori-teori permasalahan yang di ambil, Bahasa pemrograman yang digunakan, Basis data yang di gunakan dalam Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers.

3. BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas tentang Gambaran Umum Perusahaan , Analisis Fungsional, Perancangan Data, Perancangan Antar Muka., dan Perancangan Prosedural pada Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers.

4. BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas tentang :

- a. Implementasi ➔ Lingkup dan Batasan Masalah, Kebutuhan Sumber Daya, Implementasi Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers..
- b. Pengujian ➔ Lingkup dan Lingkungan, Kebutuhan Sumber Daya, Hasil Pengujian Pembuatan Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers..

5. BAB V KESIMPULAN DAN SARAN

Bab ini membahas tentang :

- a. Kesimpulan ➔ Inti dari semua isi Laporan Praktek Kerja Lapangan tentang Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers..
- b. Saran ➔ Permintaan Penulis kepada pembaca terhadap kekurangan isi dari Laporan Praktek Kerja Lapangan tentang Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance.

BAB II

LANDASAN TEORI

2.1. Teori Tentang Permasalahan

Dalam bab ini akan dijelaskan mengenai definisi dan teori-teori yang berkaitan dengan permasalahan yang ada. Berikut ini adalah teori-teori yang mendasari dari Perancangan Perangkat Lunak Pelayanan Klaim di PT. Multi Artha Insurance Brokers.

2.1.1. Sistem

Menurut Jogiyanto (dalam Yakub, 1999:1), terdapat dua kelompok pendekatan sistem di dalam mendefinisikan sistem, yaitu pendekatan pada prosedur dan pendekatan pada komponen/elemen. Pendekatan sistem pada prosedurnya mendefinisikan sistem sebagai berikut : “ Suatu sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama - sama untuk melakukan suatu kegiatan atau tujuan tertentu ”. Sedangkan pendekatan sistem pada komponen atau elemennya mendefinisikan sistem sebagai berikut : “ Sistem merupakan bagian-bagian elemen yang saling berinteraksi dan saling berhubungan untuk mencapai membentuk satu kesatuan. (Yakub, 2012).

Karakteristik Sistem

Menurut Tata Sutabri (2012:20)^[2] mendefinisikan bahwa, sistem mempunyai karakteristik sebagai berikut :

1. Komponen Sistem (*Components System*)

Suatu sistem tidak berada dalam lingkungan yang kosong, tetapi sebuah sistem berada dan berfungsi di dalam lingkungan yang berisi sistem lainnya. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Apabila suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka

akan disebut dengan *sub sistem* sedangkan sistem yang lebih besar tersebut sering disebut *supra sistem*.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya, batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar (*Environment*)

Lingkungan luar adalah apa pun di luar batas dari sistem yang dapat mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem, lingkungan luar sistem ini dapat bersifat menguntungkan ataupun yang merugikan sistem tersebut. Lingkungan luar yang menguntungkan ini tentunya harus dijaga dan dipelihara sehingga akan mendukung kelangsungan operasi sebuah sistem. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan sebuah sistem tersebut.

4. Mempunyai Penghubung (*interface*) Antar Komponen

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Penghubung inilah yang akan menjadi media yang digunakan data dari masukan (*input*) hingga keluaran (*output*). Dengan adanya penghubung, suatu subsistem dapat berinteraksi dan berintegrasi dengan subsistem yang lain membentuk suatu kesatuan.

5. Mempunyai Masukan (*input*)

Masukan atau *input* merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan pemeliharaan (*maintenance input*), yaitu bahan yang dimasukkan agar sistem tersebut dapat beroperasi dan masukan sinyal (*signal input*), yaitu masukan yang diproses untuk mendapatkan keluaran. Contoh, di dalam suatu unit sistem komputer “program”

adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan “data” adalah *signal input* untuk diolah menjadi informasi.

6. Mempunyai Keluaran (*output*)

Keluaran (*output*) merupakan hasil energi dari pemrosesan. Keluaran dapat berupa informasi yang dapat digunakan sebagai masukan untuk pengambil keputusan atau hal-hal yang menjadi *input* bagi subsistem lain atau hanya sebagai sisa pembuangan.

7. Mempunyai Pengolahan (*processing*)

Pengolahan (*process*) merupakan suatu sistem yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan. Contohnya sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Mempunyai Sasaran (*Objective*) dan Tujuan (*Goal*)

Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan (*goal*) yang bersifat *deterministic*. Apabila suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Tujuan inilah yang mengarahkan suatu sistem, tanpa adanya tujuan sistem menjadi tidak terarah dan terkendali. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

9. Mempunyai Umpan Balik (*Feed Back*)

Umpan balik diperlukan oleh bagian kendali (*Control*) sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal.

2.1.2. Informasi

Tata Sutabri (2012:29) mengemukakan pendapat bahwa, “Informasi lebih tepat disebut sebagai teori matematis dan komunikasi, sumber informasi adalah data”. Informasi adalah sebuah istilah yang tepat dalam pemakaian umum, mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi, dan lain sebagainya. Informasi juga mencakup mengenai data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Sutarman (2012:14) berpendapat bahwa, “Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.

Kualitas Informasi

Menurut Sutabri (2012:41) mendefinisikan bahwa, kualitas suatu informasi terdiri dari 3 (tiga) hal, yaitu :

1. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Informasi harus akurat karena biasanya dari sumber informasi sampai penerima informasi ada kemungkinan terjadi gangguan (*noise*) yang dapat mengubah atau merusak informasi tersebut maka informasi harus jelas mencerminkan maksudnya.

2. Tepat Waktu (*Timelines*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan. Bila pengambilan keputusan tersebut terlambat maka dapat berakibat fatal untuk organisasi.

3. Relevan (*Relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.1.3. Sistem Informasi

Sutarman (2012:13) mengemukakan pendapat bahwa, "Sistem Informasi adalah sistem yang dapat didefinisikan dengan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu. Seperti sistem lainnya, sebuah sistem informasi terdiri atas input (data, instruksi) dan output (laporan, kalkulasi)". (Sutarman, 2012)

Tata Sutabri (2012:46) berpendapat bahwa, "Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan". (Sutabri, 2012)

Berdasarkan beberapa pendapat yang dikemukakan di atas dapat ditarik kesimpulan Sistem Informasi adalah sistem yang didefinisikan dapat mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan cara mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu.

Komponen Sistem Informasi

Menurut Tata Sutabri (2012:47) mendefinisikan bahwa, suatu sistem keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran. Sistem Informasi terdiri dari beberapa komponen-komponen yang disebut blok bangunan (*building block*), diantaranya:

1. Blok Masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi, *input* yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukan yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technologi Block*)

Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Blok Teknologi terdiri dari 3 bagian utama yaitu teknisi (*Human ware* atau *Brainware*), perangkat lunak (*Software*) dan perangkat keras (*Hardware*).

5. Blok Basis data (*Database Block*)

Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidakefisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.1.4. Asuransi

Menurut RI Mehr dan Cammack dalam Danarti (2011:7) asuransi merupakan suatu lembaga untuk mengurangi risiko keuangan, dengan cara pengumpulan unit-unit exposure dalam jumlah yang memadai, untuk membuat agar kerugian individu dapat diperkiarakan. Kemudian kerugian yang dapat diramalkan itu dipikul merata oleh mereka yang bergabung.

Asuransi ialah “Perjanjian antara dua pihak atau lebih, dengan mana pihak penanggung mengikatkan diri pada Tertanggung, dengan menerima premi asuransi, untuk memberikan penggantian kepada Tertanggung karena kerugian, kerusakan atau kehilangan keuntungan yang diharapkan atau tanggung jawab hukum pihak ketiga yang mungkin akan diderita Tertanggung, yang timbul dari sebuah peristiwa yang tidak pasti, atau memberikan sebuah pembayaran yang didasarkan atas meninggal atau hidupnya seseorang yang dipertanggungkan.” (UUD Pasal 1 : 2, 1992).

2.1.5. Klaim Asuransi

Klaim asuransi adalah tuntutan dari pihak tertanggung sehubungan dengan adanya kontrak perjanjian antara asuransi dengan pihak tertanggung yang masing-masing pihak mengikatkan diri untuk menjamin pembayaran ganti rugi oleh penanggung jika pembayaran premi asuransi telah dilakukan oleh pihak tertanggung, ketika terjadi musibah yang diderita oleh pihak tertanggung.”. (Budi, 2012).

2.1.6. Data

Menurut Tata Sutabri (2012:6) mendefinisikan bahwa, “Data merupakan bahan mentah untuk diolah yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang telah diperoleh harus diukur dan dinilai baik dan buruk, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai”. (Sutabri, 2012)

Tata Sutabri (2012:72) mengemukakan pendapat bahwa, “Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata”.

Kejadian-kejadian adalah sesuatu yang terjadi saat tertentu di dalam dunia bisnis. Bisnis adalah perubahan dari suatu nilai yang disebut transaksi. Misalnya, penjualan adalah transaksi perubahan nilai barang menjadi nilai uang atau nilai piutang dagang. Kesatuan nyata adalah berupa suatu objek nyata seperti tempat, benda, dan yang betul-betul ada dan terjadi. Dari definisi dan uraian data tersebut dapat disimpulkan bahwa data adalah bahan mentah yang diproses untuk menyajikan informasi. (Sutrabi, 2012)

Berdasarkan beberapa pendapat yang dikemukakan di atas dapat ditarik kesimpulan, Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data juga merupakan bahan mentah untuk diolah yang hasilnya kemudian menjadi informasi.

2.1.7. *Unified Modelling Language (UML)*

Menurut Adi Nugroho (2010:6-7) mendefinisikan bahwa, “UML (Unified Modeling Language) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek)”. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (Nugroho, 2010)

Widodo (2011:6) mengemukakan pendapat bahwa, “UML (Unified Modelling Language) adalah bahasa pemodelan untuk sistem standar yang memiliki sintak dan semantik”. (Widodo, 2011)

Verdi Yasin (2012:194) berpendapat bahwa, “Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem”. (Yasin, 2012)

Berdasarkan beberapa pendapat yang dikemukakan di atas, maka dapat disimpulkan bahwa, UML (Unified Modeling Language) adalah bahasa pemodelan untuk visualisasi, merancang dan mendokumentasikan sistem atau

perangkat lunak yang berparadigma (berorientasi objek)". Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

2.1.8. Database

Menurut Rahardja dkk dalam jurnal CCIT (2011:238)[22], mendefinisikan bahwa, "Database adalah kumpulan fakta-fakta sebagai representasi dari dunia nyata yang saling berhubungan dan mempunyai arti tertentu". (Rahardja, 2011)

Helmi Kurniawan dan Iwan Fitrianto Rahmad mengemukakan pendapat dalam jurnal CCIT (2012:193) "*Database* atau basis data terdiri dari semua fakta yang diperlukan, dimana fakta-fakta tersebut digunakan untuk memenuhi kondisi dari kaidah-kaidah dalam sistem. Basis data menyimpan semua fakta, baik fakta awal pada saat sistem mulai beroperasi, maupun fakta-fakta yang diperoleh pada saat proses penarikan kesimpulan sedang dilaksanakan. Basis data digunakan untuk menyimpan data hasil observasi dan data lain yang dibutuhkan selama pemrosesan".

Verdi Yasin (2012:274) berpendapat bahwa, "Basis data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis, sehingga dapat digunakan oleh suatu program untuk memperoleh informasi dari basis data tersebut. Basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi". (Yasin, 2012)

Menurut Prasetyo (2012:181) mendefinisikan bahwa, "*Database* adalah sebuah struktur yang umumnya dikategorikan dalam 2 (dua) hal, sebuah *database flat* dan sebuah *database relasional*. *Database relasional* lebih disukai karena lebih masuk akal dibandingkan *database flat*". (Prasetyo, 2012)

Dari definisi ini, terdapat tiga hal yang berhubungan dengan *database*, yaitu sebagai berikut ini:

1. Data itu sendiri yang diorganisasikan dalam bentuk database.
2. Simpanan permanen (storage) untuk menyimpan database tersebut. Simpanan ini merupakan bagian dari teknologi perangkat keras yang digunakan di sistem informasi. Simpanan permanen yang umumnya digunakan berupa harddisk.
3. Perangkat lunak untuk memanipulasi database. Perangkat lunak ini dapat dibuat sendiri dengan menggunakan bahasa pemrograman komputer atau dibeli dalam bentuk suatu paket. Banyak paket perangkat lunak yang disediakan untuk memanipulasi database. Paket perangkat lunak ini disebut dengan DBMS (Database Management System).

Berdasarkan beberapa pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa, Database adalah kumpulan fakta dimana fakta tersebut digunakan untuk memenuhi kondisi dari kaidah dalam sistem. Basis data digunakan untuk menyimpan data hasil observasi dan data lain di dalam komputer secara sistematis yang dibutuhkan selama pemrosesan, sehingga dapat digunakan oleh suatu program untuk memperoleh informasi dari basis data tersebut.

2.2. Bahasa Pemrograman

Bahasa Pemrograman (*Programming language*) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa pemrograman yang lebih menekankan pada sistem ini yaitu bahasa pemrograman web sebagai berikut :

2.2.1. Website

Menurut Dina Fitria Murad, Nia Kusniawati dalam jurnal CCIT Vol. 7 No. 1 (2013:49) mendefinisikan bahwa, “*Web* adalah sistem dengan informasi

yang disajikan dalam bentuk teks, gambar, suara, dan lainnya yang tersimpan dalam sebuah *server web internet* yang disajikan dalam bentuk *hypertext*".

Rudyanto Arief (2011:7) mengemukakan pendapat bahwa, "*Web* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*". *Browser* (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. (Rudyanto, 2011)

Menurut Siti Aisyah (2012:112) mendefinisikan bahwa, "*Website* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). (Aisyah, 2012)

Berdasarkan beberapa pendapat yang dikemukakan di atas, maka dapat disimpulkan *website* adalah kumpulan halaman yang menyajikan informasi dengan berbagai macam format data seperti *text*, *image*, *animasi*, suara bahkan video yang menggunakan protokol HTTP (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser* yang dihubungkan dengan jaringan halaman (*hyperlink*) sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi.

2.2.2. PHP (PHP Hypertext Preprocessor)

Menurut Rulianto Kurniawan dalam bukunya (2010:1), "*PHP (PHP Hypertext Preprocessor)* adalah bahasa pemrograman untuk dijalankan melalui halaman *web*, umumnya digunakan untuk mengolah informasi di *internet*, sedangkan dalam pengertian lain PHP yaitu bahasa pemrograman *web server side* yang bersifat *open source*. PHP merupakan *script* yang menyatu dengan HTML dan berada pada *server (server side HTML embedded scripting)*". (Kurniawan, 2010)

Rudyanto Arief (2011:43) mengemukakan pendapat bahwa, PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*. (Rudyanto, 2011)

Anhar (2010:3) berpendapat bahwa, “PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web-server-side* yang bersifat *open source*”. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu terbaru atau *up to date*. Semua *script* dieksekusi pada *server* dimana *script* tersebut dijalankan. (Anhar, 2010)

Berdasarkan definisi diatas dapat disimpulkan bahwa PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *web server side* yang bersifat *open source* untuk dijalankan melalui halaman *web*. PHP merupakan *server side scripting* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu terbaru atau *up to date*. Semua *script* dieksekusi pada *server* dimana *script* tersebut dijalankan.

2.2.3. HTML (Hypertext Markup Language)

Menurut Rudyanto Arief (2011:23) mendefinisikan bahwa, “HTML atau *HyperText Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan pada *web browser*”. (Kurniawan, 2010)

Diari Puji (2010:13) mengemukakan pendapat bahwa, “HTML adalah bahasa yang dikenali oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*)”. (Puji, 2010)

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa, HTML adalah bahasa yang sangat tepat dipakai untuk menampilkan informasi pada halaman *web*, karena HTML menampilkan informasi dalam bentuk *hypertext* dan juga mendukung sekumpulan perintah yang dapat digunakan untuk mengatur tampilan informasi tersebut, sesuai dengan namanya, bahasa ini menggunakan tanda (*markup*) untuk menandai perintah-perintahnya.

2.2.4. JavaScript

Pada tahun 1995, Netscape mengumumkan JavaScript sebagai bahasa scripting objek yang mudah digunakan yang dirancang untuk membuat aplikasi online langsung yang menghubungkan bersama objek dan sumber daya pada kedua klien dan server. Sejak itu, telah menjadi standar untuk script sisi klien di browser Web, tetapi banyak aplikasi lain juga menyertakan mesin JavaScript. Prevalensi ini telah menyebabkan pengembang untuk menulis program besar dalam bahasa yang telah disusun untuk scripting, tetapi tidak untuk pemrograman dalam skala besar. Oleh karena itu, dukungan alat sangat diperlukan untuk membantu debug dan memelihara program-program ini. Pengembangan alat pemrograman suara yang lebih dari sekadar memeriksa sifat sintaksis membutuhkan semacam analisis program. Khususnya, analisis jenis sangat penting untuk menangkap kesalahan representasi, yang mis. membingungkan angka dengan string atau boolean dengan fungsi, di awal proses pengembangan.

J. Palsberg and Z. Su (Eds.): SAS 2009 JavaScript adalah bahasa berbasis objek yang menggunakan objek prototipe untuk model warisan. Karena hampir semua operasi yang telah ditentukan dapat diakses melalui objek prototipe, penting bahwa analisis memodelkan objek-objek ini secara tepat.

Objek adalah pemetaan dari string (nama properti) ke nilai-nilai. Secara umum, properti dapat ditambahkan dan dihapus selama eksekusi dan nama properti dapat dihitung secara dinamis.

Hasil yang tidak ditentukan, seperti mengakses properti objek yang tidak ada, diwakili oleh nilai tertentu yang tidak terdefinisi, tetapi ada perbedaan halus antara objek yang tidak memiliki properti dan objek yang memiliki properti diatur ke tidak terdefinisi.

Nilai dapat dikonversi secara bebas dari satu jenis ke tipe lainnya dengan beberapa pengecualian. Bahkan, hanya ada beberapa kasus di mana tidak ada konversi otomatis yang berlaku: nilai nol dan tidak terdefinisi tidak dapat dikonversi menjadi objek dan hanya nilai fungsi yang dapat dipanggil sebagai fungsi. Beberapa konversi otomatis tidak intuitif dan pemrogram harus menyadarinya.

Bahasa membedakan nilai-nilai primitif dan membungkus nilai-nilai primitif, yang berperilaku halus berbeda dalam keadaan tertentu.

Variabel dapat dibuat oleh tugas sederhana tanpa deklarasi eksplisit, tetapi upaya untuk membaca hasil variabel yang tidak ada dalam kesalahan runtime. Pernyataan JavaScript dengan pernyataan melanggar aturan leksikal leksikal, sehingga bahkan memecahkan nama variabel adalah tugas yang sepele.

Properti objek dapat memiliki atribut, seperti `ReadOnly`. Atribut-atribut ini tidak dapat diubah oleh program tetapi mereka harus diperhitungkan oleh analisis untuk menjaga kesehatan dan presisi.

Fungsi dapat dibuat dan dipanggil dengan jumlah parameter yang bervariasi. Objek fungsi berfungsi sebagai fungsi, metode, dan konstruktor kelas pertama dengan perilaku yang berbeda secara halus. Analisis harus menjaga penggunaan ini terpisah dan mendeteksi pola inisialisasi. Dengan fungsi eval, string yang dibangun secara dinamis dapat ditafsirkan sebagai fragmen program dan dijalankan dalam lingkup saat ini.

Bahasa termasuk fitur yang meresepkan struktur tertentu (objek global, objek aktivasi, objek argumen) dalam implementasi sistem runtime. Struktur-

struktur ini harus dimodelkan dalam suatu analisis untuk memperoleh kecukupan yang mencukupi.

2.2.5. CodeIgniter

Codeigniter adalah sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller). codeigniter bersifat free alias tidak berbayar jika anda menggunakannya. framework codeigniter di buat dengan tujuan sama seperti framework lainnya yaitu untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuat nya dari awal.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

1. Model

Model merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua intruksi yang berhubungan dengan pengolahan database di letakkan di dalam model.

2. View

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user. tampilan dari user interface di kumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

3. Controller

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya dari view kemudian controller yang mengolah intruksi.

Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di

tampilkan. jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat.

2.3. Basis Data

2.3.1. MySQL

Menurut Kurniawan Rulianto dalam Wulan Dini (2010:16) mendefinisikan bahwa, “MySQL adalah suatu jenis *database server* terkenal. MySQL termasuk jenis RDBMS (*Relational Database Management System*) yang lebih populer lewat kalangan pemograman *web*. MySQL ini mendukung bahasa pemrograman PHP”. (Kurniawan, 2010)

Antonius Nugraha (2010:10) berpendapat bahwa, “MySQL adalah suatu sistem manajemen basis data relasional RDBMS (*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan”.

Wahana Komputer (2010;111) mengemukakan pendapat bahwa, “MySQL merupakan suatu aplikasi yang digunakan untuk manajemen suatu data dan banyak digunakan khalayak diseluruh dunia”. Fungsi terpenting dari MySQL adalah sebagai *content management* suatu *website*, yaitu mengatur isi/informasi yang ditampilkan suatu *website*. (Komputer, 2010)

Berdasarkan beberapa pendapat yang dikemukakan di atas, maka dapat ditarik kesimpulan bahwa MySQL adalah jenis *database server*. MySQL termasuk jenis RDBMS (*Relational Database Management System*) yang populer dalam kalangan pemrograman *web* dan mampu bekerja dengan cepat dan kokoh. MySQL ini mendukung bahasa pemrograman PHP sehingga mudah digunakan dan banyak digunakan khalayak diseluruh dunia. Fungsi terpenting dari MySQL adalah sebagai *content management* suatu *website*, yaitu mengatur isi/informasi yang ditampilkan suatu *website*.

BAB III

ANALISIS DAN PERANCANGAN

3.1. Gambaran Umum Perusahaan

Dalam Gambaran Umum Perusahaan ini akan membahas seputar perusahaan seperti sejarah, visi misi dan struktur organisasi.

4.1.1. Profil Perusahaan

PT. MULTI ARTHA INSURANCE BROKERS (“Perusahaan”) didirikan berdasarkan Akta notaris Nomor 48 tanggal 27 juni 2005, yang dibuat di hadapan Notaris Titiek Irawati Sugianto, SH Notaris di Jakarta. Akta pendirian tersebut telah mendapat pengesahan dari Menteri Hukum dan hak Asasi Manusia Republik Indonesia dengan Surat Keputusan Nomor : C-24964 HT.01.01 TH2005 tanggal 8 September 2005.

1. Bentuk Perseroan Terbatas
2. Akte Pendirian Akta notaris Nomor 48 tanggal 27 juni 2005, yang dibuat di hadapan Notaris Titiek Irawati Sugianto, SH Notaris di Jakarta.
3. Akta pendirian tersebut telah mendapat pengesahan dari Menteri Hukum dan hak Asasi Manusia Republik Indonesiadengan Surat Keputusan Nomor : C-24964 HT.01.01 TH2005 tanggal 8 September 2005.
4. Izin Usaha dari Menteri Keuangan Republik Indonesia berdasarkan Surat Keputusan No. KEP-066/KM.5/2006 tgl 22 Februari 2006.
5. NPWP No. 02.466.850.1-011.000.
6. Tanda Daftar Perusahaan No.09.03.1.67.56399 Berlaku sampai tanggal: 5-September 2011
7. Terdaftar sebagai anggota Asosiasi Broker Asuransi Indonesia (ABAI) No. Sertifikat No : 162 / ABAI / 2009.
8. Telah memiliki Polis Professional Indemnity – PolisNo.:205.724.200.09.0109 diterbitkan oleh PT. ASURANSI JASAINDONESIA.
9. Surat Keterangan Domisili No.: 79/1.824.09 berlaku sampai dengan tanggal: 24 Maret 201.

Sesuai dengan Anggaran Dasar, Perusahaan Bergerak dalam Bidang Pialang Asuransi. Untuk menjalankan usahanya dalam bidang pialang asuransi, perusahaan telah mendapatkan izin dari Menteri Keuangan Republik Indonesia berdasarkan Surat Keputusan No. KEP-066/KM.5/2006 tanggal 22 Februari 2006

A. Visi

Menjadi perusahaan pialang asuransi yang handal di Indonesia dan mampu memberikan kontribusi positif bagi perkembangan industri asuransi di Indonesia dan Asia

B. Misi

Memberikan layanan maksimal sesuai dengan peran pialang asuransi dan benar dapat dirasakan manfaat efisien dan effectiveness oleh Nasabah.

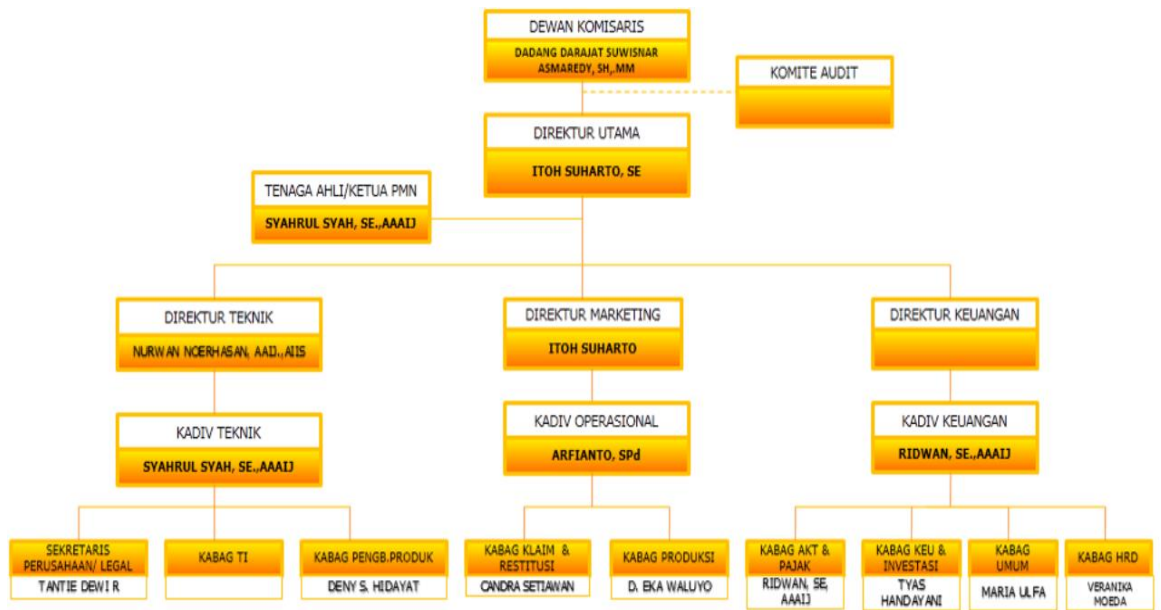
Memberikan perhatian pengetahuan pada pengembangan SDM perusahaan untuk pengembangan karir sesuai dengan tantangan industri asuransi demi peningkatan pemahaman asuransi dan manfaatnya bagi masyarakat nasional.

C. Logo Perusahaan



Gambar 3. 1 Logo Perusahaan

D. Struktur Organisasi



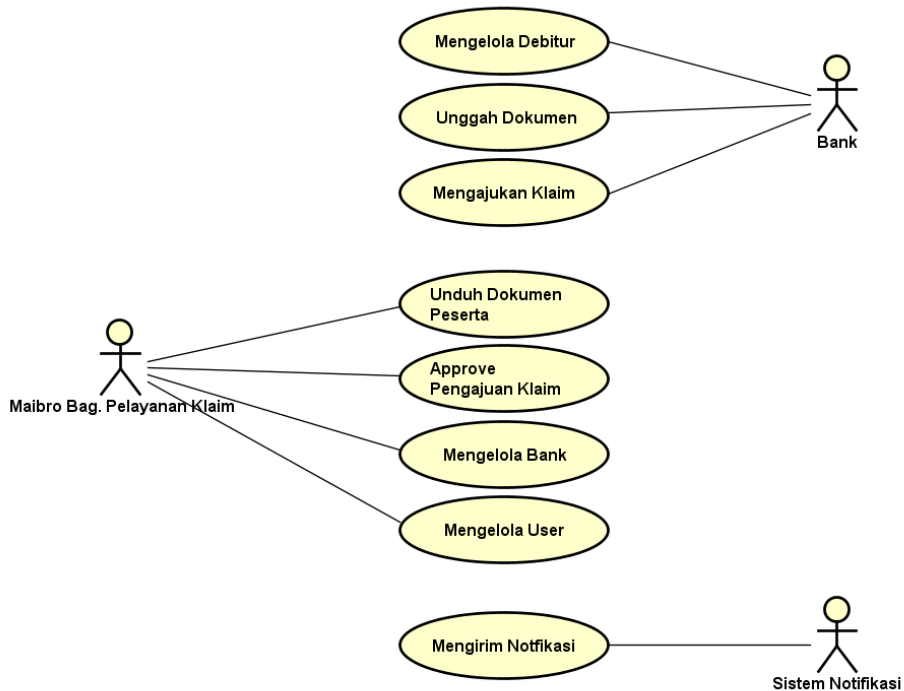
Gambar 3. 2 Stuktur Organisasi

3.2. Analisis Fungsional

Berdasarkan identifikasi persoalan yang ada, adapun gambaran umum fungsional dan fasilitas perangkat lunak yang digunakan untuk menjawab persoalan-persoalan yang dihadapi guna mencapai tujuan operasional sistem.

3.2.1. Use Case Diagram

Dari hasil analisis kebutuhan sistem maka dapat dibentuk fungsi utama yang harus ada pada sistem yang akan dimodelkan dengan menggunakan use case.



Gambar 3. 3 Use Case Diagram

3.2.2. Deskripsi Dan Skenario Use Case

1. Use case Mengelola Debitur

Nama Use Case	Mengelola Debitur	
Deskripsi Singkat	Bank menambah data baru jika ada debitur yang belum terdaftar, bank dapat mengubah dan menghapus informasi debitur	
Precondition	Bank memilih menu Debitur	
Postcondition	Data debitur tersimpan, terhapus, atau diperbaharui	
Error Situatuon	Data debitur tidak valid	
State of system error in the event	Data tidak tersimpan, terhapus atau diperbaharui dan memnunculkan pesan <i>error</i> kesalahan	
Aktor	Bank	
Trigger (Pemicu)	Tombol simpan, atau hapus di klik	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Menekan tombol tambah baru 3. Mengisi dan melengkapi <i>form</i>	2. Memunculkan <i>form</i> tambah debitur baru. 5. Memvalidasi data yang dikirim dari <i>form</i>

	4. Menekan tombol simpan	6. Menampilkan pesan berhasil 7. Menyimpan data yang dikirim dari <i>form</i>
	5. Menekan tombol ubah 7. Mengisi dan melengkapi <i>form</i> 8. Menekan tombol simpan	9. Memunculkan <i>form</i> ubah dan menampilkan data yang akan diubah. 12. Memvalidasi data yang dikirim dari <i>form</i> ubah 13. Menampilkan pesan berhasil diperbaharui 14. Menyimpan data yang diperbaharui
	1. Menekan tombol hapus 3. Menekan tombol Iya	2. Menampilkan konfirmasi hapus 4. Menampilkan pesan berhasil dihapus 5. Menghapus data yang dipilih
Skenario Alternative	1. Menekan tombol tambah baru 3. Tidak mengisi <i>form</i> dengan valid 4. Menekan tombol simpan	2. Memunculkan <i>form</i> tambah debitur baru. 5. Data yang diisi di <i>form</i> tidak valid 6. Menampilkan pesan data yang tidak valid 7. Tidak menyimpan data
	1. Menekan tombol ubah 3. Tidak mengisi <i>form</i> dengan valid 4. Menekan tombol ubah	2. Memunculkan <i>form</i> tambah debitur baru. 5. Data yang diisi di <i>form</i> tidak valid 6. Menampilkan pesan data yang tidak valid 7. Tidak memperbaharui data
	1. Menekan tombol hapus 3. Menekan tombol Iya	2. Menampilkan konfirmasi hapus 4. Menampilkan pesan berhasil dihapus 5. Menghapus data yang dipilih

Tabel 3. 1 Deskripsi Use Case Mengelola Debitur

2. Use case Unggah Dokumen

Nama Use Case	Unggah Dokumen	
Deskripsi Singkat	Bank mengunggah file dokumen debitur yang diajukan	
Precondition	Data debitur belum memiliki file dokumen	
Postcondition	Dokumen diunggah berdasarkan debiturnya	
Error Situatuon	Debitur tidak terdaftar	
State of system error in the event	Menampilkan kesalahan eror	
Aktor	Bank	
Trigger (Pemicu)	Tombol simpan diklik	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Tombol ajukan di klik 3. Memilih file yang akan diunggah 4. Menekan tombol simpan	2. Menampilkan <i>form</i> pengajuan 5. Memvalidasi ukuran dan jenis file

		6. Menyimpan file yang diunggah
Skenario Alternative		

Tabel 3. 2 Deskripsi Use Case Unggah Dokumen

3. Use case Mengajukan Klaim

Nama Use Case	Mengajukan Klaim	
Deskripsi Singkat	Bank mengajukan klaim debitur yang diajukan	
Precondition	Data debitur belum diajukan	
Postcondition	Data debitur masuk ke tabel pengajuan	
Error Situatuon	Data yang diajukan tidak valid	
State of system error in the event	Menampilkan kesalahan eror	
Aktor	Bank	
Trigger (Pemicu)	Menekan tombol pengajuan berdasarkan debiturnya	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Menekan tombol ajukan klaim 3. Mengisi form 4. Menekan tombol simpan 7. Menekan tombol kirim	2. Memunculkan form pengajuan 5. Memvalidasi yang dikirim dari form 6. Menyimpan data pengajuan 8. Mengirim data ke maibro / admin
Skenario Alternative		

Tabel 3. 3 Deskripsi Use Case Mengajukan Klaim

4. Use case Unduh Dokumen

Nama Use Case	Unduh Dokumen Peserta	
Deskripsi Singkat	Mengunduh dokumen peserta debitur yang mengajukan klaim	
Precondition	Data debitur ada pada tabel pengajuan	
Postcondition	File dokumen terunduh	
Error Situatuon	File dokumen tidak ada	
State of system error in the event	Menampilkan kesalahan eror	
Aktor	Bag. Pelayanan Klaim / admin	
Trigger (Pemicu)	Menekan tombol unduh dokumen	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Menekan tombol unduh berdasarkan debitur	2. Mengunduh file dokumen berdasarkan data yang pilih
Skenario Alternative		

Tabel 3. 4 Deskripsi Use Case Unduh Dokumen

5. Use case Memvalidasi Pengajuan Klaim

Nama Use Case	Memvalidasi Pengajuan Klaim	
Deskripsi Singkat	Memeriksa data dan file yang diajukan jika valid maka statusnya akan diubah, jika tidak maka akan dikembalikan ke bank dengan keterangan	
Precondition	Data debitur ada pada tabel pengajuan	
Postcondition	Merubah status pengajuan debitur	
Error Situatuon	Data debitur tidak memiliki dokumen	
State of system error in the event	Menampilkan kesalahan eror dan tidak memprosesnya	
Aktor	Bag. Pelayanan Klaim / admin	
Trigger (Pemicu)	Menekan tombol <i>approve</i>	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Menekan <i>aprove</i> berdasarkan debitur	2. Memperbaharui status pengajuan
Skenario Alternative		

Tabel 3. 5 Deskripsi Use Case Memvalidasi Pengajuan Klaim

5. Use case Mengelola Bank

Nama Use Case	Mengelola Bank	
Deskripsi Singkat	Admin menambah data bank baru, admin dapat mengubah dan menghapus informasi bank	
Precondition	Admin memilih menu bank	
Postcondition	Data bank tersimpan, terhapus, atau diperbaharui	
Error Situatuon	Data bank tidak valid	
State of system error in the event	Data tidak tersimpan, terhapus atau diperbaharui dan memunculkan pesan <i>error</i> kesalahan	
Aktor	Bag. Pelayanan Klaim / Admin	
Trigger (Pemicu)	Tombol simpan, atau hapus di klik	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Menekan tombol tambah baru 3. Mengisi dan melengkapi <i>form</i> 4. Menekan tombol simpan	2. Memunculkan <i>form</i> tambah debitur baru. 5. Memvalidasi data yang dikirim dari <i>form</i> 6. Menampilkan pesan berhasil 7. Menyimpan data yang dikirim dari <i>form</i>
	1. Menekan tombol ubah 3. Mengisi dan melengkapi <i>form</i> 4. Menekan tombol simpan	2. Memunculkan <i>form</i> ubah dan menampilkan data yang akan diubah. 5. Memvalidasi data yang dikirim dari <i>form</i> ubah 6. Menampilkan pesan berhasil diperbaharui

		7. Menyimpan data yang diperbaharui
	1. Menekan tombol hapus 3. Menekan tombol Iya	2. Menampilkan konfirmasi hapus 4. Menampilkan pesan berhasil dihapus 5. Menghapus data yang dipilih
Skenario Alternative	1. Menekan tombol tambah baru 3. Tidak mengisi <i>form</i> dengan valid 4. Menekan tombol simpan	2. Memunculkan <i>form</i> tambah debitur baru. 5. Data yang diisi di <i>form</i> tidak valid 6. Menampilkan pesan data yang tidak valid 7. Tidak menyimpan data
	1. Menekan tombol ubah 3. Tidak mengisi <i>form</i> dengan valid 4. Menekan tombol ubah	2. Memunculkan form tambah debitur baru. 5. Data yang diisi di form tidak valid 6. Menampilkan pesan data yang tidak valid 7. Tidak memperbaharui data
	1. Menekan tombol hapus 3. Menekan tombol Iya	2. Menampilkan konfirmasi hapus 4. Menampilkan pesan berhasil dihapus 5. Menghapus data yang dipilih

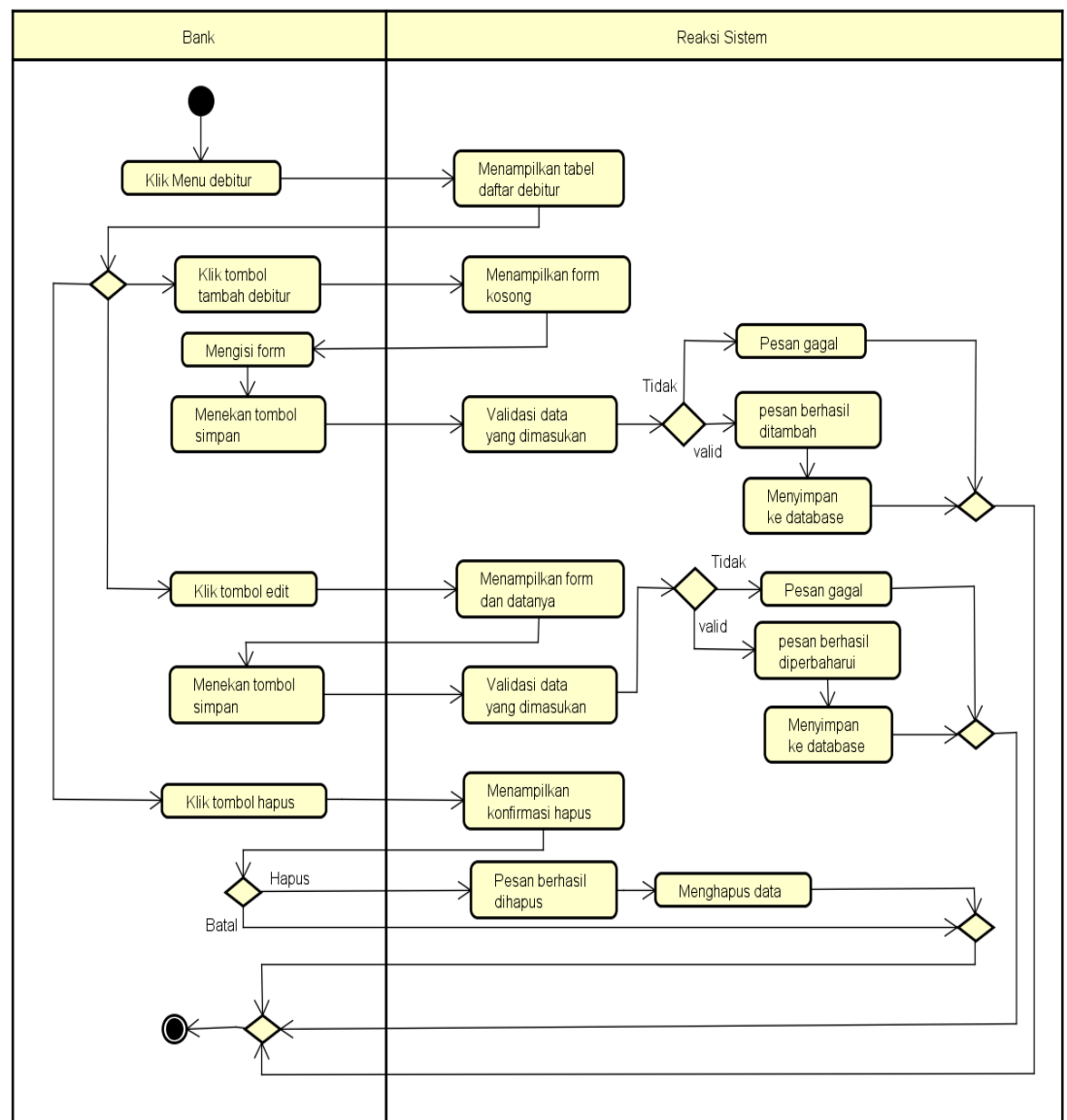
Tabel 3. 6 Deskripsi Use Case Mengelola Bank

5. Use case Mengirim Notifikasi

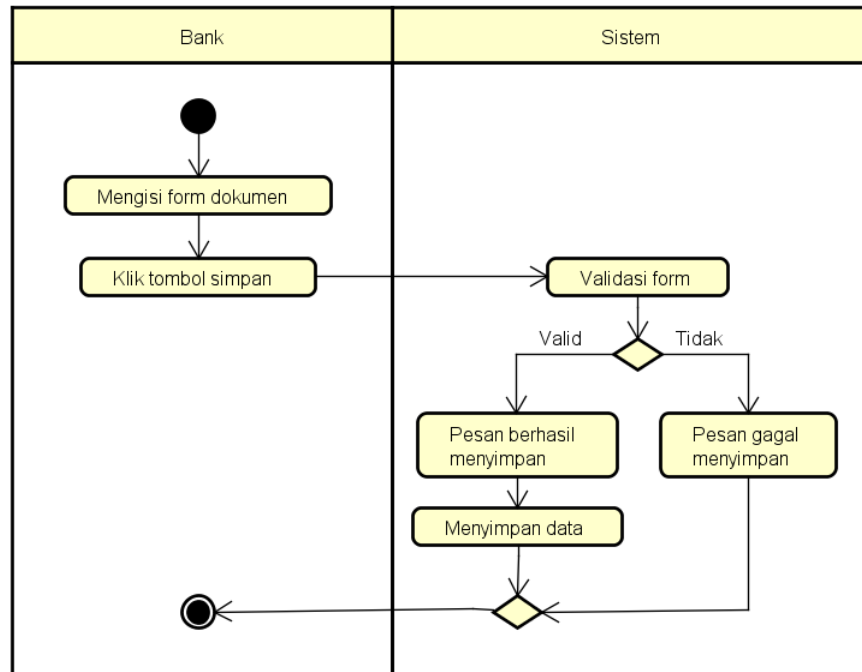
Nama Use Case	Mengirim Notifikasi	
Deskripsi Singkat	Mengirimkan notifikasi pada setiap aksi usecase dan memberikan keterangannya	
Precondition	Tidak ada permintaan notifikasi	
Postcondition	Menampilkan notifikasi serta keterangan	
Error Situatuon	Permintaan notif tidak valid	
State of system error in the event	Notifikasi tidak dikirimkan	
Aktor	Sistem notifikasi	
Trigger (Pemicu)	Permintaan data notifikasi	
Skenario Normal	Aksi Aktor	Reaksi Sistem
	1. Mengirimkan data notifikasi	2. Menerima data notifikasi 3. Menampilkan data notifikasi
Skenario Alternative		

Tabel 3. 7 Deskripsi Use Case Mengirimkan Notifikasi

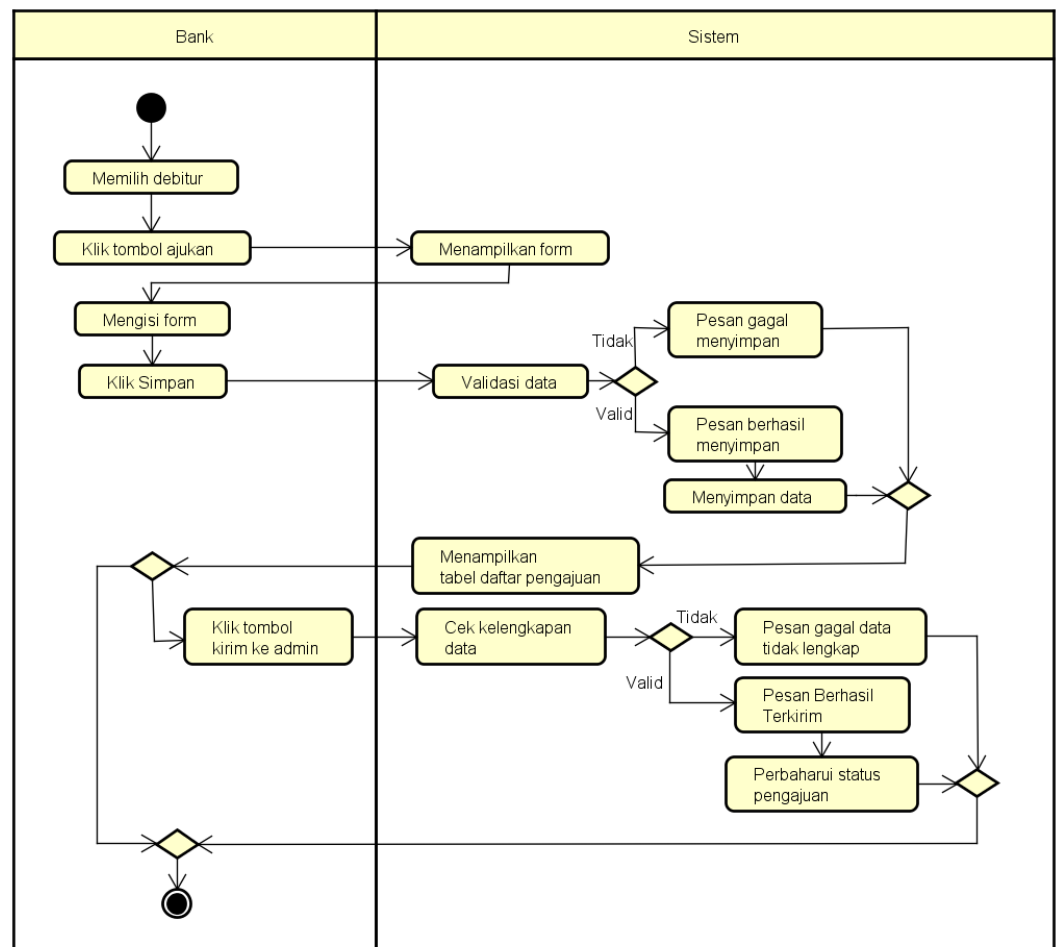
3.2.3. Diagram Aktivitas (Activiti Diagrams)



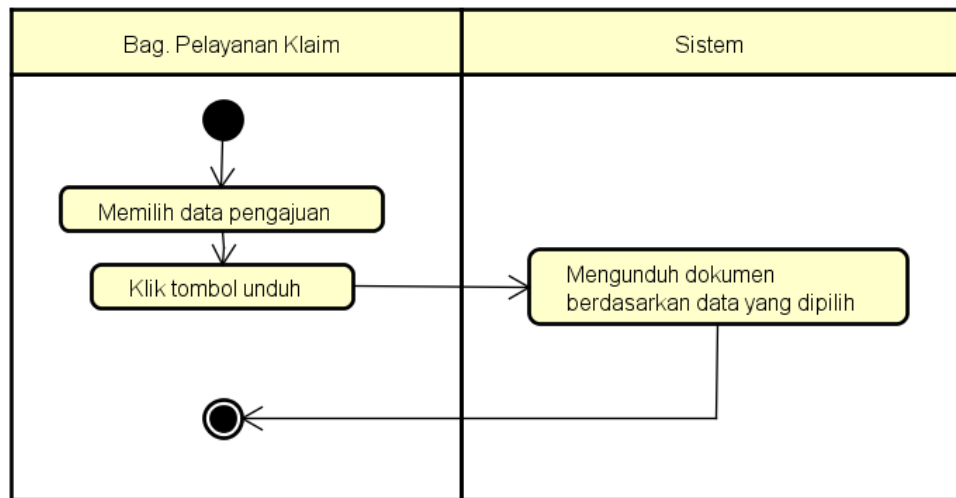
Gambar 3. 4 Activity Diagram Mengelola Debitur



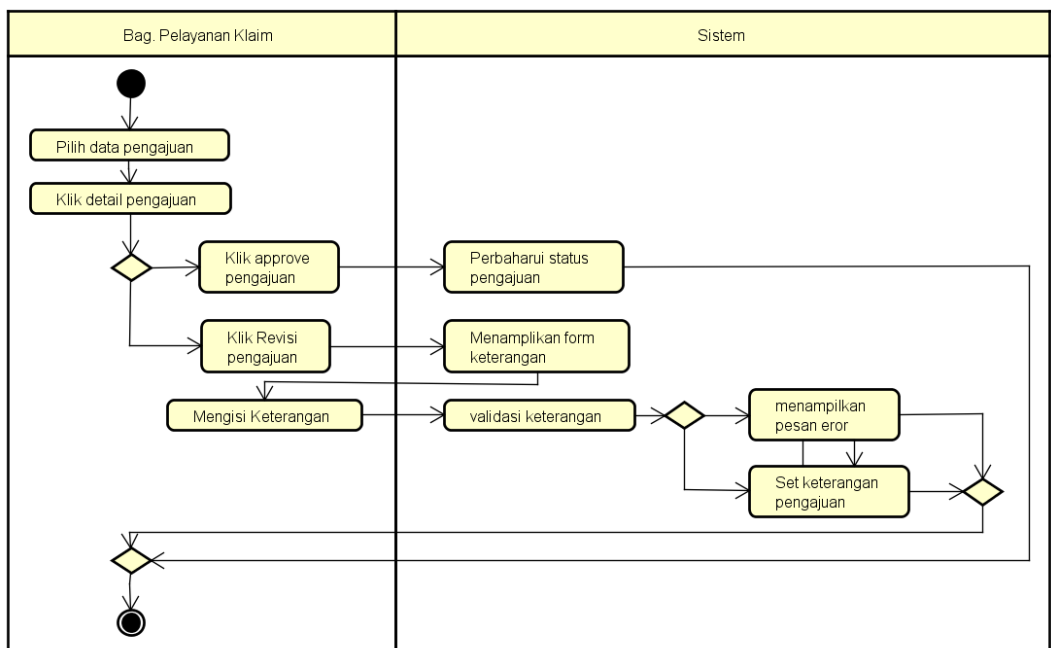
Gambar 3. 5 Activity Diagram Unggah Dokumen



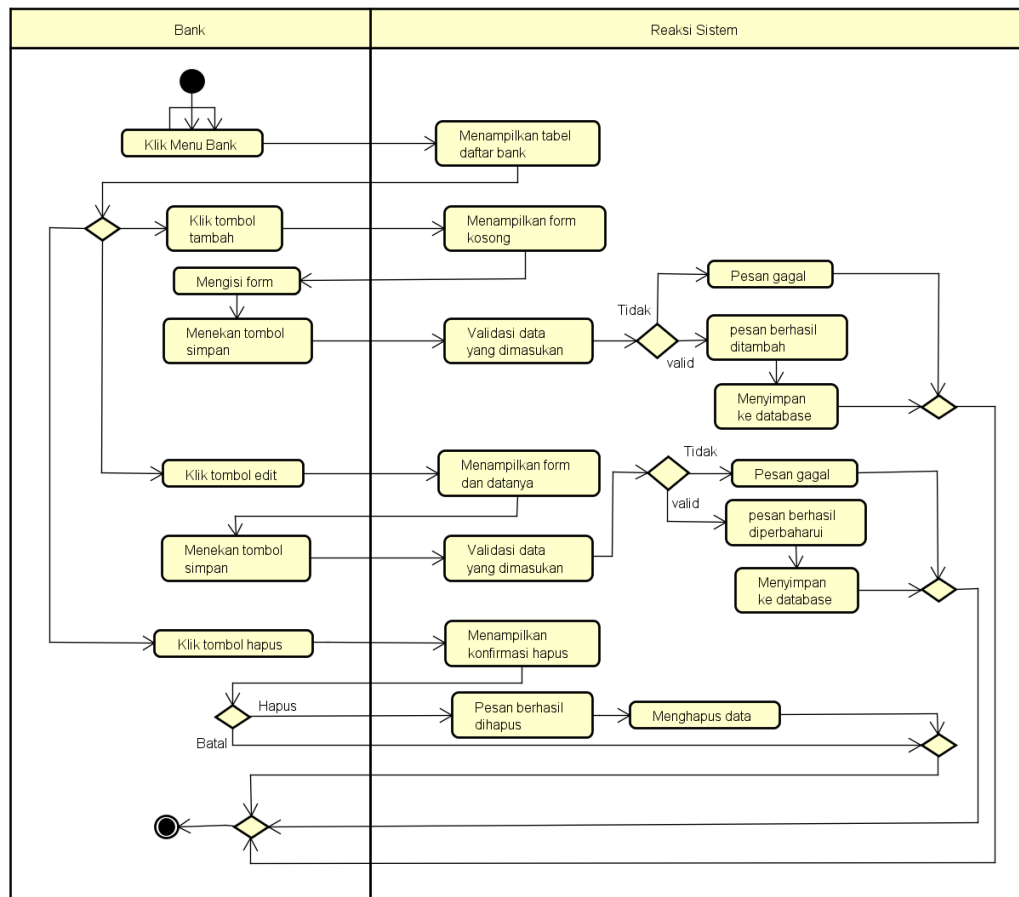
Gambar 3. 6 Activity Diagram Mengajukan Klaim



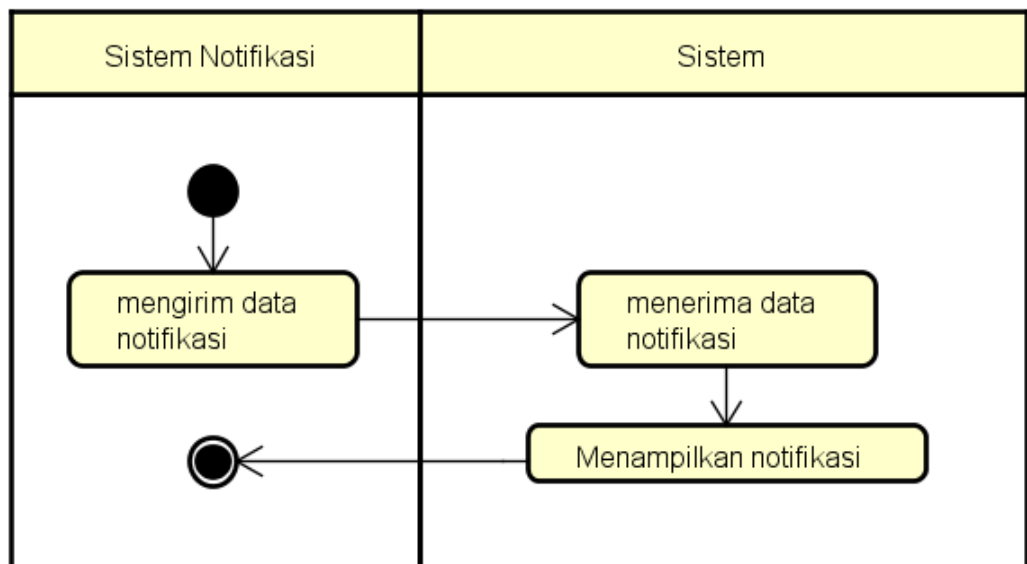
Gambar 3. 7 Activity Diagram Unduh Dokumen peserta



Gambar 3. 8 Activity Diagram Validasi Pengajuan

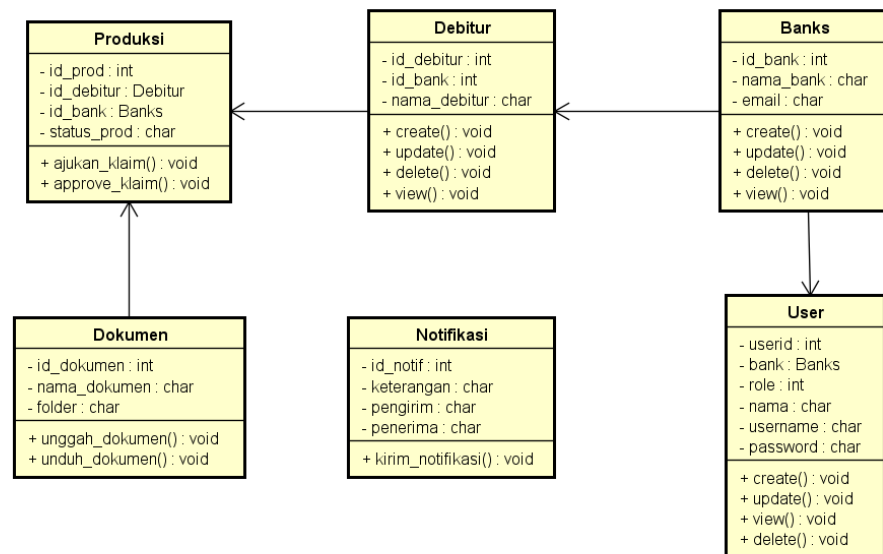


Gambar 3. 10 Activity Diagram Kelola Bank



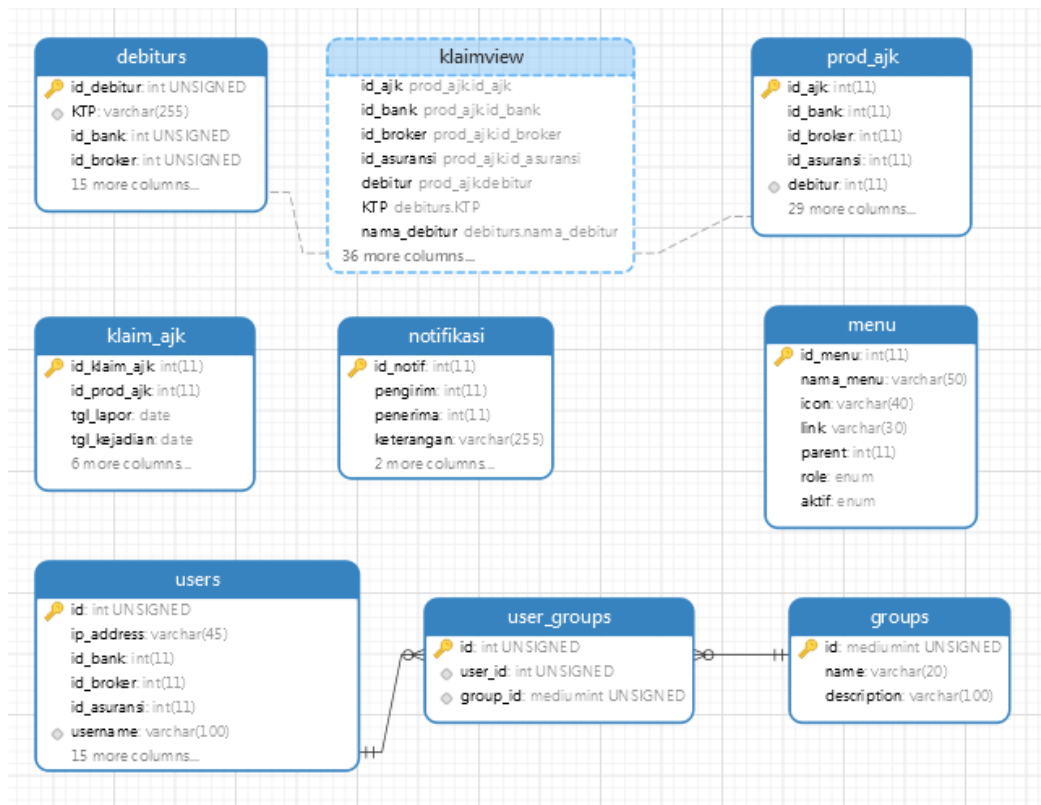
Gambar 3. 9Activity Diagram Mengirim Notifikasi

3.2.4. Diagram Kelas (*Class Diagrams*)



Gambar 3. 11 Class Diagram

3.3. Perancangan Data

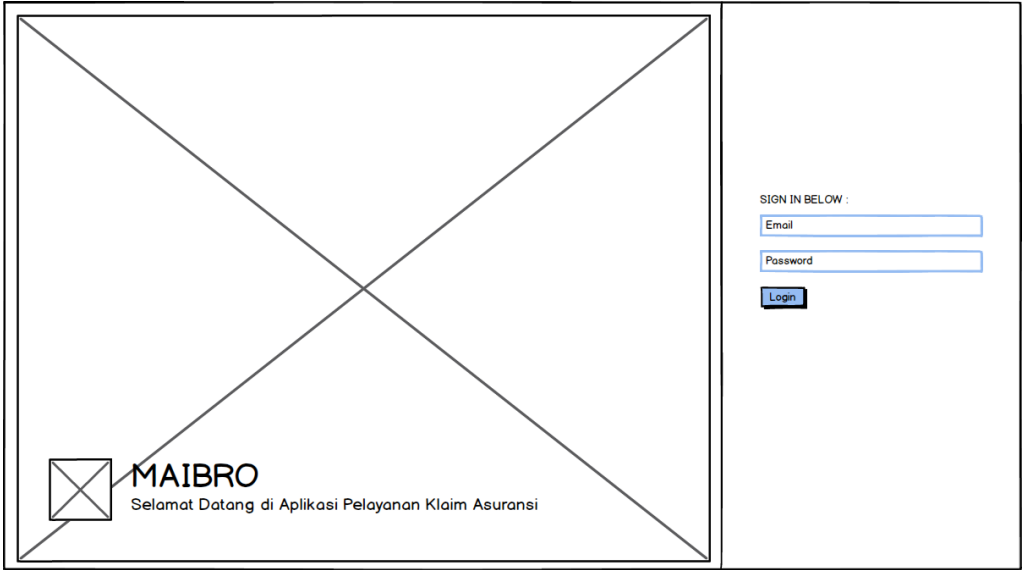


Gambar 3. 12 Perancangan data

3.4. Perancangan Antarmuka

1. Halaman Login

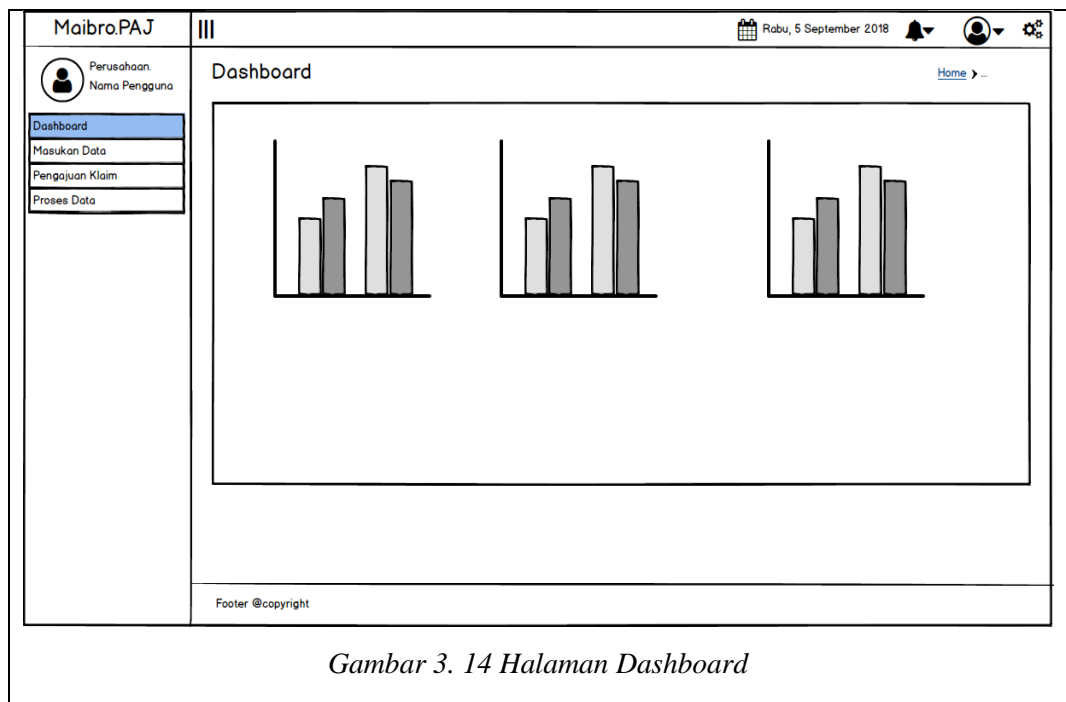
Nama antarmuka	:	Halaman login
Fungsi	:	Untuk menjaga keamanan dari pengguna lain dan memsiahkan halaman antar pengguna
Layout antarmuka	:	



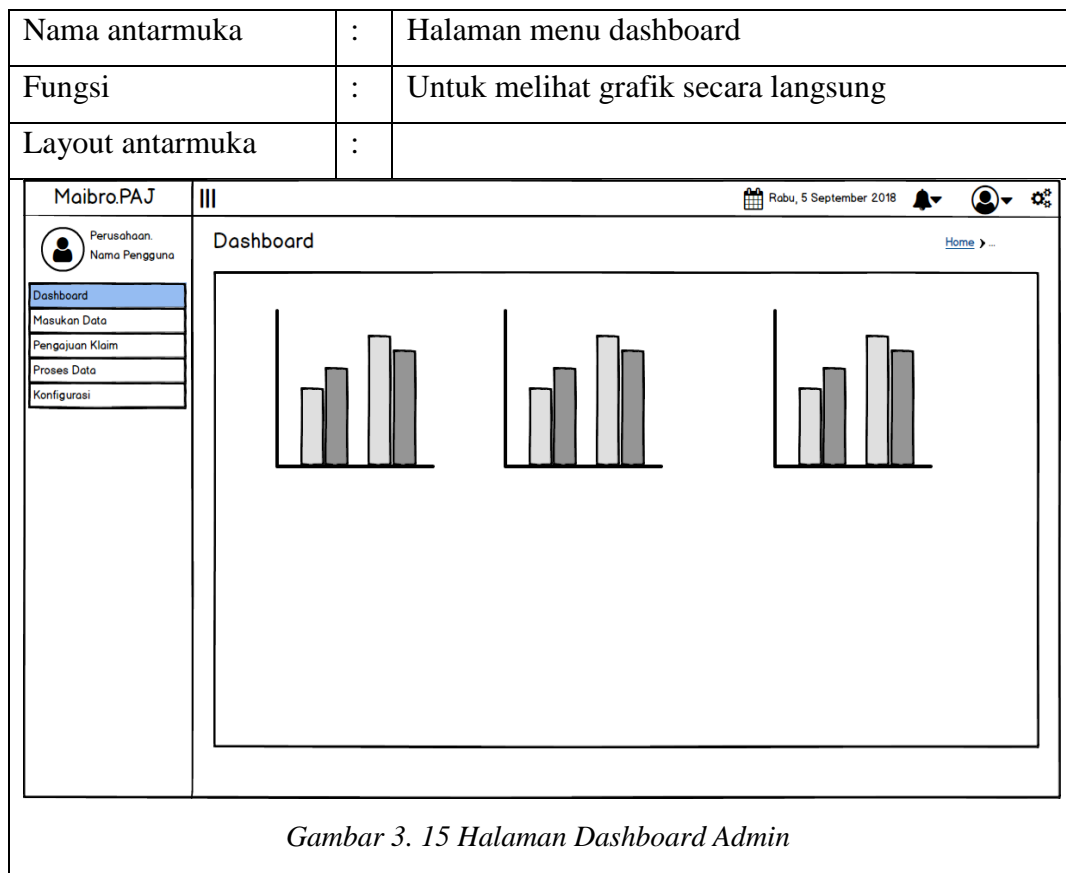
Gambar 3. 13 Halaman Login

2. Halaman Dashboard Bank

Nama antarmuka	:	Halaman Dashboard
Fungsi	:	Untuk melihat grafik
Layout antarmuka	:	



3. Halaman Dashboard Admin



4. Halaman Menu Debitur

Nama antarmuka	:	Debitur
Fungsi	:	Untuk menambah, memperbaharui, atau menghapus data debitur
Layout antarmuka	:	

Gambar 3. 16 Halaman Menu Debitur

5. Halaman form debitur

Nama antarmuka	:	Form Tambah Debitur
Fungsi	:	Form masukan untuk menambah atau mengedit data debitur
Layout antarmuka	:	

7. Halaman Form Pengajuan Klaim

Nama antarmuka	:	Halaman Form Pengajuan Klaim
Fungsi	:	Untuk melengkapi persyaratan dokumen
Layout antarmuka	:	

The screenshot shows the 'Form Pengajuan Klaim' (Claim Submission Form) in the Maibro.PAJ system. The interface is divided into a sidebar menu on the left and a main form area. The sidebar menu includes options like 'Dashboard', 'Masukan Data', 'Pengajuan Klaim', 'Klaim Jiwa', and 'Proses Data'. The main form area contains several sections for data entry:

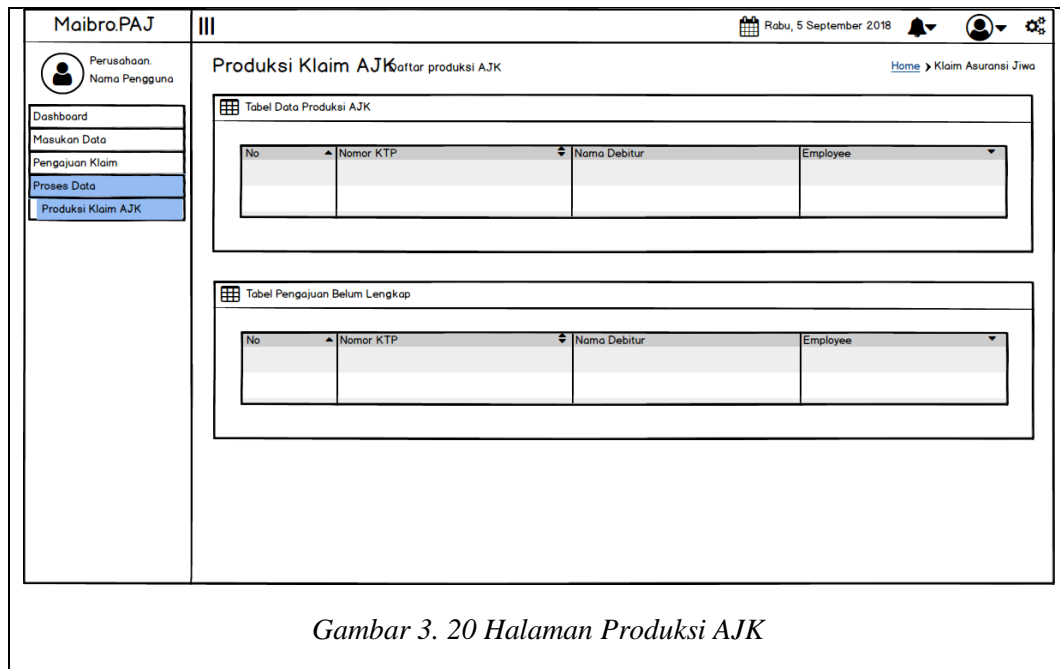
- Personal Information:** Fields for Nomor KTP (13312233442212), Nama Debitur (Rizki Pebrianto), Email (rizkipebrianto96@gmail.com), Jenis Kelamin (Pria), Tanggal Lahir (5 Februari 1996), Umur (22), Telepon (081315189696), and Alamat (Komp. Bumi Asri Blok H.7).
- Credit Information:** Fields for Nomor Kredit (Masukan Nomor Kredit), Masa Kredit (Masukan Masa Kredit), Plafon Pinjaman (Masukan Plafon Pinjaman), Premi (Masukan Premi), Jumlah Klaim (Masukan Jumlah Klaim), and Jenis Kredit (- Pilih -).
- Document Uploads:** Sections for Surat Kepesertaan (e-policy), Surat Keterangan Kematian, Foto KTP Identitas Debitur, Surat Pengajuan Klaim, Catatan / Historical Angsuran, and Photo Saat Akad Kredit. Each section has a 'Pilih File' button and a message 'Tidak ada file yang dipilih'.
- Actions:** Buttons for 'Simpan', 'Batal', and 'Kirim Ke Admin'.

The top of the interface shows the date 'Rabu, 5 September 2018' and a navigation breadcrumb: 'Home > Asuransi Jiwa > Form Pengajuan Klaim'.

Gambar 3. 19 Form Pengajuan Klaim

8. Halaman Proses Data AJK

Nama antarmuka	:	Halaman Produksi AJK
Fungsi	:	Untuk memonitor pengajuan serta pengiriman pengajuan ke admin
Layout antarmuka	:	



3.5. Perancangan Prosedural

Adapun perancangan procedural dari sistem ini adalah sebagai berikut :

1. Halaman login

Input	:	email & password
Output	:	Info invalid username, password
Bentuk	:	Lihat Gambar 3.13 Halaman Login
Algoritma	:	
<pre> if email=true & password=true then If role=admin then Dashboard admin, Menu => admin Menu => member else Dashboard bank, Menu => member end if end if </pre>		

2. Halaman menu debitur

Nama Modul	:	Isi tabel debitur
Kegunaan	:	Untuk memisahkan data debitur antar bank
Input	:	-
Output	:	Data tabel berdasarkan bank
Bentuk	:	Gambar 3. 16 Halaman Menu Debitur
Algoritma	:	
If id_bank = idbank_user Select * from debitur where id_bank = id_bankuser endif		

3. Halaman form debitur

Nama Modul	:	<i>Insert</i> debitur baru
Kegunaan	:	Menambahkan data debitur baru
Input	:	Masukan dari form
Output	:	Berhasil menyimpan di tabel
Bentuk	:	Gambar 3. 16 Halaman Menu Debitur
Algoritma	:	
If data[] = true INSERT INTO debitur VALUES(data[]) Else Pesan -> data tidak valid endif		

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Perangkat Lunak

Implementasi program Perangkat Lunak Pelayanan Asuransi Jiwa Kredit Untuk Mempermudah Pengajuan Klaim di Departemen Pelayanan Kalim PT. Muli Artha *Insurance Brokers* dilakukan dengan menggunakan metode Black Box Testing. Metode Black Box Testing merupakan pengujian program yang mengutamakan pengujian terhadap kebutuhan fungsi dari suatu program. Tujuan dari metode Black Box Testing ini adalah untuk menemukan kesalahan fungsi pada program.

4.1.1. Lingkup dan Batasan

Pada implementasi atau pada tahapan penerapan adalah suatu bentuk kegiatan yang merupakan rangkaian lanjutan dari kegiatan perancangan perangkat lunak. Implementasi dimaksudkan sebagai usaha untuk mewujudkan hasil dari perancangan perangkat lunak. Hasil ini berarti hasil dari kegiatan rangkaian implementasi perancangan perangkat lunak adalah proses aplikasi yang telah dibuat dan sudah dapat digunakan serta berfungsi dengan baik.

Adapun penjelasan mengenai ruang lingkup dan batasan implementasi, termasuk modul program yang akan diimplementasikan, fungsional sistem dan lain sebagainya termasuk lingkungan implementasi adalah sebagai berikut :

1. Aplikasi yang dihasilkan berupa aplikasi berbasis web dengan menggunakan framework CodeIgniter.
2. Aplikasi hanya dapat menyimpan

4.1.2. Spesifikasi Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dan yang dibutuhkan oleh aplikasi atau sistem adalah suatu unit personal komputer. Perangkat keras yang diusulkan ini dibuat berdasarkan kebutuhan sistem saat ini danantisipasi kebutuhan di masa yang akan datang. Konfigurasi yang digunakan pada desain sistem yang diusulkan adalah

Processor	: Intel Core i3 @ 2.1Ghz
Monitor	: 17" HD LED
Mouse	: Optic
RAM	: Minimum 2 GB
Harddisk	: Minimum 500 GB
Keyboard	: <i>Classic</i>
VGA	: Minimum <i>On Board</i>
Printer	: Laserjet (jika dibutuhkan)

4.1.3. Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan adalah aplikasi berdasarkan kebutuhan sistem. Dalam pembuatan program dan desain sistem menggunakan berbagai software. Adapun Konfigurasi sistem software yang digunakan sebagai berikut:

Sistem Operasi	: Windows 7 atau lebih
Web browser	: Mozilla firefox, Chrome, Windows explorer
<i>AppServ</i>	: PHP dan MySQL
Aplikasi	: PDF <i>Viewer</i> , Microsoft Office

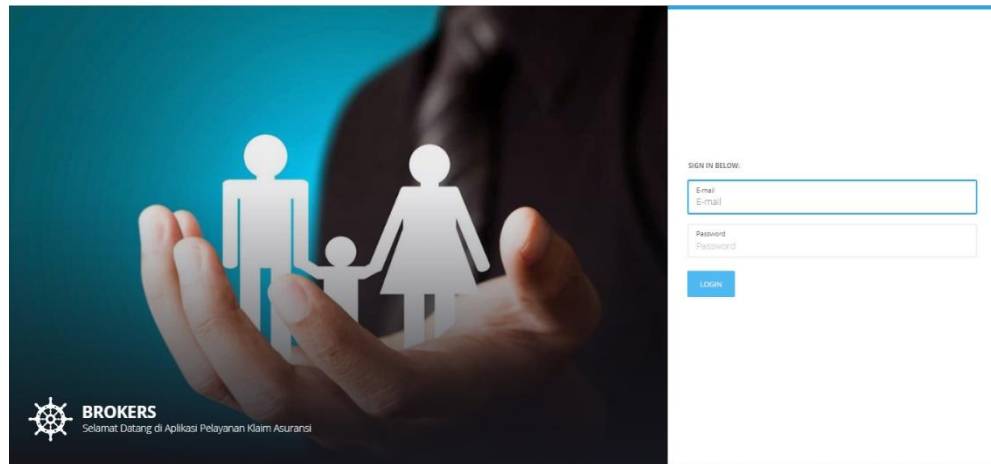
4.1.4. Hak Akses (*Brainware*)

Terdapat 2 actor yang dapat meng-akses sistem ini, yaitu :

1. Bag. Pelayanan Asuransi (Sebagai Administrator)
2. Bank / Klien perusahaan Maibro

4.1.5. Implementasi Antarmuka

1. Antarmuka login



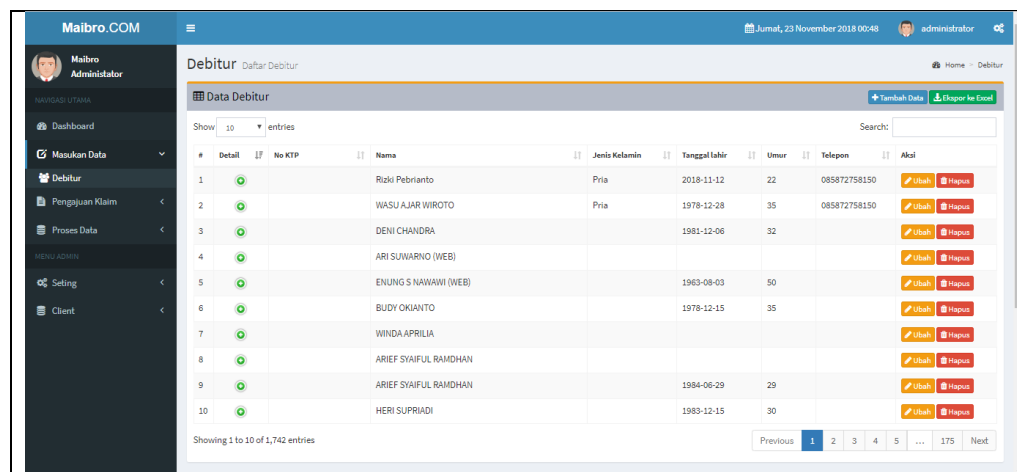
Gambar 4. 1 Antarmuka Login

Keterangan : Halaman login aplikasi

Uraian cara penggunaan :

1. Masukkan email dan password
2. Tekan tombol *login in*

2. Antarmuka menu Debitur



Keterangan : Halaman index menu debitur

Fungsi : Mengelola data debitur di masing masing bank

3. Antarmuka menu Klaim Asuransi Jiwa

#	Detail	KTP	Nama	Jenis Kelamin	Tanggal lahir	Umur	Telepon	Aksi
1			ALFIAN MUJANI		1960-05-03	54		Ajukan Klaim
2			WINDA APRILIA		1988-04-12	26		Ajukan Klaim
3			MUJIANAH		1982-02-27	32		Ajukan Klaim
4			ADE CANDRA HIDAYATULLOH		1988-12-03	25		Ajukan Klaim
5			IMAM HIDAYAT		1983-09-08	30		Ajukan Klaim
6			FITRI YULIANA SIRINGO RINGO		1983-07-12	30		Ajukan Klaim
7			NOVIANA PUSPITA DEWI (MORTGAGE)		1984-11-25	29		Ajukan Klaim
8			YANE SOPHIANA DEWI		1981-01-07	33		Ajukan Klaim
9			SUKMA WILJAYA		1976-10-26	37		Ajukan Klaim
10			RUDI HARTONO		1982-09-15	31		Ajukan Klaim

Keterangan : Halaman index pengajuan klaim

Fungsi : Untuk mengajukan klaim dan mengisi form dokumen bagi bank

4. Antarmuka nemu proses data pengajuan AJK

#	Detail	Nomor Kredit	Debitur	SPK	EP	SKK	KTP	HA	AK	Status Dok	Aksi
1		123123123	DIDIN (WEB)	✓	✓	✓	✓	✓	✓	Sudah Lengkap	Update Kirim ke admin
2		123	RIO ROBIN	-	✓	-	-	-	-	Belum Lengkap	Update Kirim ke admin
3		123113	RIO ROBIN	-	✓	-	-	-	-	Belum Lengkap	Update Kirim ke admin
4		1222	ALFIAN MUJANI	✓	-	-	-	-	-	Belum Lengkap	Update Kirim ke admin
5		12	WAWAN HERMAWAN	✓	-	-	-	-	-	Belum Lengkap	Update Kirim ke admin
6		112	NOVIANA PUSPITA DEWI (MORTGAGE)	✓	-	-	-	-	-	Belum Lengkap	Update Kirim ke admin
7			FATIMAH AZZAHRA	✓	-	-	-	-	-	Belum Lengkap	Update Kirim ke admin

#	Detail	No Sertifikat	Nomor Kredit	Debitur	SPK	EP	SKK	KTP	HA	AK	Status Dok	Keterangan
1			123123	SUPRIHATININGSIH (WEB)	✓	✓	✓	✓	✓	✓	Sudah Lengkap	
2			1233	MUHAMAD RAFAI	✓	✓	✓	✓	✓	✓	Sudah Lengkap	
3			12	AMIDAH	✓	✓	✓	✓	✓	✓	Sudah Lengkap	

Keterangan : Halaman index proses data pengajuan klaim

Fungsi : Untuk mengirimkan debitur yang diajukan dengan persyaratan dokumen harus dilengkapi

5. Antarmuka form tambah data debitur

The screenshot shows the 'Form Debitur' interface in the Maibro.COM application. The left sidebar contains navigation links: Dashboard, Masukan Data, Debitur, Pengajuan Klaim, and Proses Data. The main form area is titled 'Debitur Form Debitur' and includes the following fields:

- Nomor KTP: Input field with placeholder 'Masukan Nomor KTP'.
- Jenis Kelamin: Dropdown menu with placeholder '-- Pilih --'.
- Nama Debitur: Input field with placeholder 'Masukan Debitur'.
- Tempat Lahir: Input field with placeholder 'Masukan Kota Kelahiran'.
- Email: Input field with placeholder 'Masukan Email'.
- Nomor Telepon: Input field with placeholder 'Masukan Nomor Telepon'.
- Tanggal Lahir: Date picker with placeholder 'Pilih Tanggal'.
- Umur: Input field with placeholder 'Masukan Umur'.
- Alamat: Text area with placeholder 'Masukan Alamat'.

At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). The footer of the page displays 'Copyright © 2014-2016 Almasaeed Studio. All rights reserved.' and 'Version 2.4.0'.

Keterangan : Halaman form tambah data debitur

Fungsi : Untuk melengkapi informasi data debitur yang ditambahkan

6. Antarmuka edit data informasi debitur

The screenshot shows the 'Form Debitur' interface in the Maibro.COM application, used for editing existing data. The left sidebar is identical to the previous screenshot. The main form area is titled 'Debitur Form Debitur' and contains the same fields as before, but with pre-filled data:

- Nomor KTP: Input field with placeholder 'Masukan Nomor KTP'.
- Jenis Kelamin: Dropdown menu with 'Pria' selected.
- Nama Debitur: Input field with 'WASU AJAR WIROTO' entered.
- Tempat Lahir: Input field with placeholder 'Masukan Kota Kelahiran'.
- Email: Input field with 'WASU.AJARWIROTO@bjb.co.id' entered.
- Nomor Telepon: Input field with '085872758130' entered.
- Tanggal Lahir: Date picker with '1978-12-28' selected.
- Umur: Input field with '35' entered.
- Alamat: Text area with 'Komp. Bumi Asri blok H.7 Rt 04 Rw 09 Kel. Gempol S' entered.

At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). The footer of the page displays 'Copyright © 2014-2016 Almasaeed Studio. All rights reserved.' and 'Version 2.4.0'.

Keterangan : Halaman form edit data debitur

Fungsi : Untuk merubah informasi data debitur yang dipilih

7. Halaman form pengajuan klaim

Keterangan : Halaman form pengajuan klaim asuransi jiwa kredit










Fungsi : Untuk melengkapi informasi debitur yang diajukan serta melengkapi form dokumen persyaratan

4.2. Pengujian

Pengujian dengan metode Black Box Testing dilakukan dengan cara memberikan sejumlah input pada program. Input tersebut kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi dapat menghasilkan output yang sesuai dengan yang diinginkan dan sesuai pula dengan fungsi dasar dari program tersebut. Apabila dari input yang diberikan proses dapat menghasilkan output yang sesuai dengan kebutuhan fungsionalnya, maka program yang dibuat sudah benar, tetapi apabila output yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program tersebut, dan selanjutnya dilakukan penelusuran perbaikan untuk memperbaiki kesalahan yang terjadi.

4.2.1. Pengujian Blax Box Pada Menu Login

No	Skenario Pengujian	Hasil Yang Di Harapkan	Kesimpulan
1	Mengosongkan username dan password, lalu klik tombol login	Tombol login tidak akan berfungsi	Valid

			
2	<p>Hanya mengisi username, lalu tekan tombol login</p> 	<p>Sistem tidak akan merespon dan tombol tidak akan berfungsi</p> 	Valid
3	<p>Hanya mengisi password, lalu tekan tombol login</p> 	<p>Sistem tidak akan merespon dan tombol tidak akan berfungsi</p> 	Valid
4	<p>Menghasilkan dengan salah satu data benar, lalu klik login</p> 	<p>Sistem akan menolak akses login dan menampilkan pesan error</p> 	Valid
5	<p>Menghasilkan username dan password dengan benar lalu klik login</p> 	<p>Menuju ke halaman dashboard</p> 	Valid

4.2.2. Pengujian *Blax Box* Pada Menu Data Debitur

No	Skenario Pengujian	Hasil Yang Di Harapkan	Kesimpulan
1	Validasi masukan yang ditulis didalam form 	Tombol login tidak akan berfungsi 	Valid
2	Hanya mengisi username, lalu tekan tombol login 	Sistem tidak akan merespon dan tombol tidak akan berfungsi 	Valid
3	Hanya mengisi password, lalu tekan tombol login 	Sistem tidak akan merespon dan tombol tidak akan berfungsi 	Valid
4	Menghasilkan dengan salahsatu data benar, lalu klik login 	Sistem akan menolak akses login dan menampilkan pesan error 	Valid
5	Menghasilkan username dan password dengan benar lalu klik login 	Menuju ke halaman dashboard 	Valid

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Selama praktek kerja lapangan di PT. Multi Artha Insurance Brokers, penulis dapat menarik suatu kesimpulan yaitu sebagai berikut:

1. Pemilahan dokumen dokumen pengajuan menjadi lebih efektif dikarenakan dokumen disimpan di sistem.
2. Proses pencarian data debitur menjadi lebih mudah dan efisien.
3. Menghilangkan biaya operasional jasa pengiriman dokumen.

5.2. Saran

Adapun saran yang dapat penulis berikan untuk pengembangan dari aplikasi yang telah dibuat antara lain :

1. Sistem ini dapat dikembangkan sampai menuju proses pembayaran asuransi
2. Sebaiknya ada sistem backup pada database agar data keamanan data menjadi lebih terjamin dan untuk mengantisipasi jika sistem mengalami crash.

DAFTAR PUSAKA

- Aisyah, S. (2012). *Aplikasi Pengajuan Kredit Berbasis Web pada PT.Adira Quantum Multifinance*. Tangerang : Perguruan Tinggi Raharja. Jurnal CCIT Vol.5 No. 2.
- Anhar. (2010). *Panduan Menguasai PHP & MySQL Secara Otodidak*. Jakarta : Media Kita.
- Komputer, W. (2010). *Shourtcourse SQL Server 2008 Express*. Yogyakarta : Andi.
- Kurniawan, R. (2010). *PHP & MySQL Untuk Orang Awam*. Palembang: Maxikom.
- Nugroho, A. (2010). *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.
- Prasetio, A. (2012). *Buku Pintar Pemrograman Web*. Jakarta: Mediakita.
- Puji, D. (2010). *Menjadi Programmer Jempolan Menggunakan PHP*. Yogyakarta: Mediakom.
- Rahardja, U. (2011). *Peningkatan Kinerja Distributed Database Melalui Metode DMQ Base Level*. Tangerang: Perguruan Tinggi Raharja. Jurnal CCIT Vol.4 No.3. Mei 2011.
- Rudyanto, A. (2011). *Pemrograman Web Dinamis Menggunakan PHP & MySQL*. Yogyakarta : Andi.
- Sutanta. (2004). Informasi.
- Sutarman. (2012). *Definisi Sistem*. Yogyakarta: Andi Offset.
- Sutrabi, T. (2012). *Sistem Informasi Manajemen*. Yogyakarta: Andi Ofset.
- UUD Pasal 1 : 2. (1992). *Tentang Usaha Perasuransian*. Jakarta: Republik Indonesia.
- Widodo. (2011). *Menggunakan UML Unified Model Language*. Bandung: Informatika.
- Yakub. (2012). *Pengantar Sistem*. Yogyakarta: Graha Ilmu.
- Yasin, V. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek Pemodelan, Arsitektur dan Perancangan*. Jakarta: Mitra Wacana Media.