



Analisis Contingency Plan Sistem Billing Import Menggunakan Metode User Centered Design

Penulis :

Ilham Dwi Prasetyo Nugroho

Rd. Nuraini Siti Fatonah, S.S., M.Hum

D4 Teknik Informatika, Politeknik Pos Indonesia

Jalan Sari Asih No.54, Kota Bandung, Jawa Barat

Judul :

Analisis *Contingency Plan* Sistem *Billing Import* Menggunakan Metode User
Centered Design

2021,

... Halaman, 16 cm x 23 cm

Penulis :

Ilham Dwi Prasetyo Nugroho

Rd. Nuraini Siti Fatonah, S.S., M.Hum, SFPC

Penyunting : M.Harry K Saputra, S.T., M.T.I, SFPC

Layout : Ilham Dwi Prasetyo Nugroho

Sanksi Pelanggaran Pasal 113

Undang-Undang Nomor 28 Tahun 2014

tentang Hak Cipta

- (1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
- (2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1)

huruf c, huruf d, huruf t dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

- (3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
- (4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

KATA SAMBUTAN

Kata Pengantar

Puji dan syukur kehadiran Tuhan YME atas rihmat dan kenikmatan kepada penulis untuk membuat buku ini dari hasil internship yang dilakukan pada salah satu terminal petikemas di Jakarta. Penulis berharap semoga buku ini dapat menambah pengetahuan pengalaman bagi para pembaca sehingga kedepannya dapat memperbaiki maupun menambah isi buku ini agar lebih baik lagi

Penulis yakin masih banyak kekurangan yang terdapat didalam buku ini, karena keterbatasan pengetahuan dan pengalaman penulis. Oleh karena itu, penulis sangat mengharapkan kritikan ataupun masukan dari para pembaca yang membangun demi kesempurnaan laporan ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga buku ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Bekasi, 29 Januari 2022

Penulis

Daftar Isi

D4 TEKNIK INFORMATIKA, POLITEKNIK POS INDONESIA	II
KATA SAMBUTAN.....	IV
KATA PENGANTAR	V
DAFTAR ISI.....	VI
DAFTAR GAMBAR.....	VIII
DAFTAR SIMBOL.....	X
DAFTAR TABEL	XV
BAB I.....	1
1.1 PENGENALAN EKPOR IMPOR.....	1
1.3 OBJEK PERMASALAHAN	2
1.3 TOPIK PERMASALAHAN.....	4
1.4 TUJUAN, MANFAAT DAN RUANG LINGKUP.....	4
BAB II.....	6
2.1 SISTEM.....	8
2.1.1 <i>Karakteristik Sistem</i>	8
2.2 IMPOR	9
2.3 UML.....	10
2.3.1 <i>Jenis Jenis UML</i>	11
2.4 DATABASE	13
2.5 E-BILLING	14
2.6 MySQL	15
2.7 PHP.....	15
2.8 CONTINGENCY PLAN	15
2.8.1 <i>Elemen Contingency Plan</i>	16
2.9 METODE UCD(<i>USER CENTERED DESIGN</i>)	17
2.9.1 <i>Bagan dari Metode Ucer Centered Design (UCD)</i>	18
2.10 KUESIONER	19
2.11 SONARLINT	19
BAB III.....	22
3.1 TAHAPAN DIAGRAM ALUR METODOLOGI PENELITIAN	23
3.1.1 <i>Analisis Kebutuhan</i>	23
3.1.2 <i>Desain Sistem</i>	25


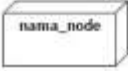

3.1.3	<i>Implementasi</i>	26
3.1.4	<i>Integrasi dan Pengujian</i>	27
3.1.5	<i>Pemeliharaan</i>	27
3.2	PERUMUSAN METODE <i>UCER CENTERED DESIGN</i>	28
3.2.1	<i>Perumusan Usability Testing</i>	30
3.2.2	<i>Contoh Perhitungan Usability Testing</i>	31
BAB IV		33
4.1	<i>FLOWCHART</i> SISTEM YANG SEDANG BERJALAN.....	33
4.2	<i>FLOWCHART</i> SISTEM YANG AKAN DIBANGUN.....	34
4.2.1	<i>Admin</i>	34
4.2.2	<i>User</i>	36
4.3	PERANCANGAN.....	37
4.3.1	<i>Usecase Diagram</i>	37
4.3.2	<i>Class Diagram</i>	47
4.3.3	<i>Sequence Diagram</i>	48
4.3.4	<i>Perancangan Database</i>	54
4.3.5	<i>Perancangan Interface</i>	55
4.3.6	<i>Perancangan Perangkat Lunak dan Keras</i>	62
4.4	PENGUJIAN.....	64
4.4.1	<i>White Box</i>	64
4.4.2	<i>Usability Testing</i>	78
KESIMPULAN		80
DAFTAR PUSTAKA		81






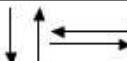
Daftar Gambar

Gambar 1-2.1 <i>Contingency Plan Hierarchies</i>	16
Gambar 2-2.2 Proses <i>User Centered Design</i>	18
Gambar 3-2.3 <i>Sonarlint</i>	20
Gambar 4-2.4 Contoh kesalahan terdeteksi.....	20
Gambar 5-2.5 Contoh Tidak Terjadi Kesalahan.....	21
Gambar 6-3.1 Diagram Alur Metodologi Penelitian.....	23
Gambar 7-4.1 Flowchart yang berjalan	33
Gambar 8-4.2 Flowchart yang dibangun	34
Gambar 9-4.3 Flowmap User	36
Gambar 10-4.4 Usecase Diagram.....	38
Gambar 11-4.5 Class Diagram sistem survey <i>contingency plan</i> sistem <i>billing</i>	48
Gambar 12-4.6 Sequence Diagram Admin <i>Login</i>	49
Gambar 13-4.7 Sequence Diagram Menejemen Kategori Pertanyaan	50
Gambar 14-4.8 Sequence Diagram Menejemen Pertanyaan	51
Gambar 15-4.9 Sequence Hasil Kuisisioner	52
Gambar 16-4.10 Sequence Diagram User.....	53
Gambar 17-4.11 Perancangan Database.....	54
Gambar 18-4.12 Halaman Utama User	55
Gambar 19-4.13 Halaman Kuesioner	56
Gambar 20-4.14 Halaman Login.....	57
Gambar 21-4.15 Dashboard Admin	58
Gambar 22-4.16 HalamanManajemen Kategori	58
Gambar 23-4.17 Halaman Manajemen Pertanyaan	59
Gambar 24-4.18 Halaman Hasil Kuisisioner	60
Gambar 25-4.19 Halama Hasil Per kategori	61
Gambar 26-4.20 Halaman Laporan.....	62
Gambar 27-4.21 Pengujian Halaman Utama	65

Gambar 28-4.22 Pengujian Koneksi Database	66
Gambar 29-4.23 Pengujian aksi Kuesioner	67
Gambar 30-4.24 Pengujian Validasi	68
Gambar 31-4.25 Pengujian Logout	69
Gambar 32-4.26 Pengujian gantipass.....	70
Gambar 33-4.27 Pengujian Menu Home.....	71
Gambar 34-4.28 Pengujian Menu Kategori	72
Gambar 35-4.29 Pengujian <i>CRUD</i> Kategori	73
Gambar 36-4.30 Pengujian Menu Pertanyaan	74
Gambar 37-4.31 Pengujian <i>CRUD</i> Pertanyaan	75
Gambar 38-4.32 Pengujian Menu Hasil	76
Gambar 39-4.33 Pengujian Hasil Responden.....	77
Gambar 40-4.34 Pengujian Submenu Laporan.....	78
Gambar 41-4.35 penerapan Usability Testing Data Keseluruhan	79
Gambar 42-4.36 penerapan Usability Testing data per grup	79

















Daftar Simbol



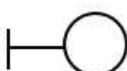



Simbol	Deskripsi
Package 	package merupakan sebuah bungkus dari satu atau lebih node
Node 	biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang dilukissertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan / dependency 	Kebergantungan antar node, arah panah mengarah pada node yang dipakai
Link 	relasi antar node



NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya





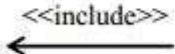
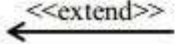
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan 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.

Analisis Contingency Plan Sistem Billing Impor Menggunakan Metode User Centered Design

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Daftar Tabel

Table 1-2,1 State of the art contingency plan	7
Table 2-3.1 Tabel hasil wawancara	25
Table 3-3.2 Pertanyaan Kuisisioner Usability Testing sistem	28
Table 4-3.3 Pilihan jawaban Usability Testing	29
Table 5-3.4 Kategori Hasil Penilaian	31
Table 6-3.5 Contoh Responden <i>usability testing</i>	31
Table 7-3.6 Tabel setelah dilakukan Aturan <i>Usability Testing</i>	32
Table 8-4.1 Definisi Aktor	38
Table 9-4.2 Definisi Usecase	39
Table 10-4.3 Skenario Usecase Login	40
Table 11-4.4 skenario Kelola kategori pertanyaan	41
Table 12-4.5 skenario kelola pertanyaan	43
Table 13-4.6 Skenario hasil kuesioner	44
Table 14-4.7 Skenario pengisian kuesioner	45
Table 15-4.8 <i>Hardware</i>	63
Table 16-4.9 <i>Software</i>	63

BAB I

Pendahuluan

1.1 Pengenalan Ekpor Impor

Transaksi perdagangan luar negeri yang sering disebut ekspor impor hakikatnya merupakan transaksi sederhana dan tidak lebih dari jual atau beli barang perusahaan dengan perusahaan yang berada dinegara berbeda. Namun, pertukaran barang maupun jasa yang menyeberangi laut ataupun darat itu tidak jarang timbul permasalahan yang kompleks antara pengusaha yang mempunyai kebudayaan, bahasa dan adat yang berbeda-beda(Roselyne Hutabarat, 1997).

Ekspor dan impor di suatu negara merupakan kegiatan dari perdagangan internasional yang sering terjadi untuk memenuhi kebutuhan yang tidak bisa di produksi di negara tersebut. Proses impor dan ekspor bisa berjalan karena ada penghubung antara importir maupun ekportir dengan pihak jasa pengiriman seperti pelabuhan. Menurut Ahmad Ahyar dalam situs <https://arahkata.pikiran-rakyat.com/ekbis/pr-128995357/kemenperin-nilai-peran-pelabuhan-sangat-penting-dalam-perdagangan-internasional>, “Kementerian Perindustrian menilai peran pelabuhan sangat penting dalam perdagangan internasional, karena sebagai pintu gerbang utama dalam proses bongkar muat ekspor dan impor.”

Kota Jakarta memiliki banyak Pelabuhan yang bisa digunakan untuk kegiatan impor maupun ekspor, dan merupakan tempat strategis serta merupakan jalur perdagangan yang masuk ke Indonesia. Salah satu pelabuhan yang terbesar dan terkenal di Jakarta adalah Pelabuhan Tanjung Priok. Daerah Tanjung Priok sendiri ada beberapa perusahaan yang bergerak dibidang jasa pengiriman ekspor maupun impor, diantaranya : PT. Jakarta International Container Terminal, KSO. Terminal Petikemas Koja, PT. Multi Terminal Indonesia, serta yang baru dibuat untuk menambah terminal di Jakarta yaitu PT. New Port Container Terminal 1.

1.3 Objek Permasalahan

Terminal Petikemas X merupakan salah satu perusahaan jasa dibidang pelayanan ekspor dan impor yang terdapat di Daerah Jakarta Utara. Terminal Petikemas X sendiri bukan merupakan perusahaan perseroan terbatas melainkan perusahaan dalam bentuk Kerja Sama Operasi. Perusahaan ini dibentuk untuk membantu lancarnya proses ekspor dan impor yang dikhawatirkan, terminal Y tidak bisa menampung lonjakan arus barang container ekspor dan impor yang akan terjadi. Terminal Petikemas X sendiri akan menciptakan smart digital terminal yang mengembangkan inovasi digital. Inovasi digital yang dimaksud adalah Pelayanan *Billing* yang semula masih dilakukan secara manual *Billing*, sekarang sudah menjadi pelayanan *Billing* mandiri melalui fasilitas *E-Billing*, pembayaran melalui media elektronik

dengan memanfaatkan kode billing sebagai kode transaksi (Dewi, Pratiwi, et all, 2019), serta pembayaran yang menggunakan EDC dari beberapa bank yang sudah melakukan kerjasama *Host to Host*(H2H).

E-Billing diharapkan bisa mempercepat transaksi sehingga transaksi dapat berjalan tanpa ada antrian ketika melakukan pembayaran. Tetapi, Ada salah satu kekurangan di dalam *E-Billing* tersebut yaitu ketika ada masalah di *server* tempat *E-Billing* beroperasi, transaksi akan terhambat dan bisa menjadi penumpukan pelanggan dalam melakukan pembayaran karena menunggu *server E-Billing* tersebut kembali normal. Untuk mengantisipasi hal tersebut terjadi, pihak perusahaan menyediakan alternatif lainnya berupa *contingency plan*.

Contingency Plan atau *plan* darurat merupakan langkah yang diambil ketika *server* di *e-billing* utama mengalami gangguan sehingga terjadinya penghambatan transaksi para importir, *Contingency Plan* yang akan di bangun merupakan sebuah aplikasi *billing* offline di perusahaan Terminal Petikemas X, ketika *e-billing* sudah kembali normal, data data yang ada di aplikasi *billing* offline akan bisa diinputkan ke *server* utama *e-billing* ketika sudah kembali normal.

Sebelum terciptanya *contingency plan*, terlebih dahulu dilakukan kuisisioner terhadap *user* yang akan menggunakan sistem dari *contingency plan* tersebut. Kuisisioner berfungsi untuk mengetahui apakah sebuah *contingency plan* dibutuhkan di sebuah perusahaan atau tidak. Kuesisioner tersebut menerapkan *User Centered Design* dan

Usability Testing untuk menentukan contingency plan tersebut. *Usability testing* menjadi faktor penting dari suatu sistem yang bisa dijadikan ukuran sistem yang digunakan pengguna untuk mencapai tujuan pembuatan sistem yang efektif, efisien dan penuh kepuasan dari segi konteks pengguna (Riftika et all, 2019).

1.3 Topik Permasalahan

Setelah uraian objek permasalahan yang terdapat di terminal petikemas X. Adapun topik permasalahan yang diangkat dalam buku ini, yaitu:

1. Bagaimana untuk mengetahui *contingency plan* diperlukan dalam sebuah perusahaan?
2. Bagaimana aplikasi survey *contingency plan* dibangun di Terminal Petikemas X?

Dari topik topik diatas, nantinya kita akan mengetahui tujuan dan manfaat sebuah *contingency plan* yang harus tersedia di perusahaan besar terutama yang bergerak dibidang jasa pengiriman.

1.4 Tujuan, Manfaat dan Ruang Lingkup

Berdasarkan topik permasalahan yang sudah dijelaskan di atas, adapun tujuan, manfaat dan ruang lingkup akan dijelaskan berikut ini:

Tujuan dari topik permasalahan di atas adalah :

1. Mengetahui seberapa dibutuhkan *contingency plan* dalam sebuah perusahaan.
2. Mengetahui aplikasi survey *contingency plan* yang dibangun di Terminal Petikemas X.

Sedangkan manfaat yang didapat oleh penulis yaitu dengan adanya laporan ini, bisa menambah ilmu penulis laporan maupun pembaca tentang bagaimana antisipasi atau *plan* darurat ketika terjadi sesuatu di *server billing* utama.

Ruang lingkup dari penelitian ini adalah analisis *contingency plan* sistem *Billing* di Terminal Petikemas X hanya dibagian impor dan menegetahui seberapa penting contingency plan dalam sebuah perusahaan.

BAB II

Persiapan Analisis dan Perancangan

Bab ini merupakan landasan landasan teori untuk menunjang dari penelitian ini. Teori teori didapat dari berbagai sumber jurnal, buku maupun situs yang terdapat di *website*. Adapun *state of the art* atau penelitian sebelumnya akan dicantumkan sebelum membahas tentang teori yang dibutuhkan dalam penelitian ini, berikut penelitian yang sudah ada mengenai *contingency plan* :

Table 1-2,1 State of the art contingency plan

no	Judul Penelitian	nama Peneliti	Metode penelitian	Hasil Penelitian	Perbedaan dari penelitian saya
1	Perancangan Contingency Planning Disaster Recovery Unit Teknologi Informasi Perguruan Tinggi menggunakan NIST SP800-34	Rima Dias Ramadhani, Wahyu Adi Prabowo	Water fall	1. Hasil dari analisa risk assessment TI yang dilakukan di Institut Teknologi Telkom Purwokerto dengan menggunakan NIST Sp 800-34 dihasilkan beberapa resiko terhadap infrastruktur TI dan Sistem TI yang dapat dijadikan sebuah acuan terhadap penerapan rencana pemulihan sistem jika terjadi suatu bencana 2. Dari analisa value chain terdapat 10 kritikal aset Sistem TI yang terdapat di Institut Teknologi Telkom Purwokerto	1. perbedaan dari segi metode penelitian, saya menggunakan metode User Centered Design, 2. hasil yang dihasilkan berbeda
2	PENERAPAN DISASTER RECOVERY AND CONTINGENCY PLANNING PADA PERLINDUNGAN ARSIP VITAL DI BPN DIY	Lastria Nurtanzil	Kualitatif	menghasilkan Konsep DRCP mengenalkan tentang 3 tahapan penanganan bencana yaitu: tahapan pre bencana, pada saat bencana dan pasca bencana.	1. dari object yang diteliti 2. dari metode yang digunakan

Tabel ini merupakan beberapa jurnal yang membahas tentang *contingency plan*. Pada sub-bab selanjutnya, akan membahas tentang teori-teori yang digunakan pada penelitian ini.

2.1 Sistem

Menurut Fatansyah (2015:11) “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu. “.

2.1.1 Karakteristik Sistem

Untuk penjelasan yang lebih detail, membangun serta mengembangkan sesuatu sistem yang benar, maka perlu ada yang membedakan antara unsur-unsur dari sistem yang terbentuk, seperti *Components, Boundary, Environment, Input, Output, Interfaces, Proses serta Objectives dan Goal* (Hanif Al Fatta, 2007 :5).

a. Komponen Sistem (*Components*)

Proses sistem yang mengirimkan input menjadi bentuk setengah jadi berupa output.

b. Batasan Sistem (*Boundary*)

Menggambarkan suatu unsur yang termasuk dalam sistem dan yang bukan termasuk dalam sistem atau di luar sistem.

c. Lingkungan (*Environment*)

Lingkungan merupakan unsur yang berada di luar sistem. Lingkungan juga menyediakan asumsi, kendala serta inputan terhadap suatu sistem.

d. Masukan Sistem (*Input*)

Inputan merupakan sumber daya (data, energi, peralatan) yang dibutuhkan dan dimanipulasi oleh suatu sistem.

e. Keluaran Sistem (*Output*)

Output merupakan sumber daya (informasi, laporan, dokumen, tampilan layar maupun barang jadi) yang telah berubah dari proses input di dalam suatu sistem.

f. Penghubung (*Interface*)

Interface merupakan tempat dimana pengguna sistem melakukan interaksi terhadap sistem.

g. Pengolah Sistem(*Process*)

Sistem mempunyai satu bagian yaitu untuk pengolahan data dari input yang akan dikeluarkan menjadi output.

h. Sasaran Sistem (*Objective*)

Sistem pasti akan memiliki tujuan dari terbentuknya sistem tersebut. Dari tujuan tersebut, sangat menentukan masukan yang dibutuhkan oleh sistem dan akan mengeluarkan outputan yang sesuai dengan tujuan sistem tersebut.

2.2 Impor

Impor adalah proses pembelian barang atau jasa asing dari suatu negara ke negara lain. Tingkat impor dipengaruhi oleh hambatan peraturan perdagangan (Nopirin 2011). Pemerintah

mengenakan tarif (pajak) pada produk impor, pajak itu biasanya dibayar langsung oleh importir, yang kemudian akan membebankan kepada konsumen berupa harga lebih tinggi dari produknya, ketika pemerintah asing menerapkan tarif, kemampuan perusahaan asing untuk bersaing di negara-negara itu dibatasi. Pemerintah juga dapat menerapkan kuota pada produk impor, yang membatasi jumlah produk yang dapat di impor (BR Silitonga, et all, 2019). Berdasarkan penjelasan dari sumber yang didapat, kesimpulannya adalah impor merupakan suatu proses pembelian barang yang berasal dari negara lain yang dikenakan pajak oleh pemerintah sehingga kebutuhan barang atau jasa didalam negeri terpenuhi.

2.3 UML

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma objek oriented.

UML adalah salah satu *tool* / model untuk merancang pengembangan *software* yang berbasis *object oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas

dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*.

2.3.1 Jenis Jenis UML

2.3.1.1 *Usecase Diagram*

Usecase diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dengan kata lain, *use case diagram* digunakan untuk mengetahui fungsi-fungsi apa saja yang terdapat di dalam sistem dan siapa saja yang berhak mengakses fungsi tersebut.

2.3.1.2 *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabelvariabel yang dimiliki oleh suatu kelas. Metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

2.3.1.3 *Sequence Diagram*

Sebuah Diagram yang diperuntukan untuk mendeskripsikan dan menampilkan interaksi antara objek-objek dalam sebuah system secara terperinci. selain itu, system

menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya.

2.3.1.4 *Activity Diagram*

Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu system. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu system dan memberi tekanan pada aliran kendali antar objek.

2.3.1.5 *Statechart Diagram*

Statechart atau biasa disebut dengan state diagram digunakan untuk mendokumentasikan beragam kondisi atau keadaan yang terjadi terhadap sebuah class dan kegiatan apa saja yang dapat merubah keadaan/kondisi tersebut. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

2.3.1.6 *Component Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke

dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

2.3.1.7 *Deployment Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run time*). Dengan ini memuat simpul-simpul (*node*) beserta komponen-komponen yang ada di dalamnya. Deployment diagram berhubungan erat dengan diagram kompoen dimana deployment diagram memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

2.4 *Database*

Menurut Indrajani (2015:70), “basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi”. *Database* (basis data) atau dengan sebutan pangkalan data ialah suatu kumpulan sebuah informasi yang disimpan didalam sebuah perangkat komputer secara sistematis sehingga dapat diperiksa dengan menggunakan suatu program komputer agar dapat informasi dari basis data tersebut.

2.5 *E-Billing*

E-Billing adalah pembayaran melalui media elektronik dengan memanfaatkan kode *billing* sebagai kode transaksi. *E-Billing* diharapkan bisa mempercepat transaksi sehingga transaksi bisa berjalan tanpa ada antrian ketika melakukan pembayaran (Citra Dewi, et al, 2019). *E-billing* merupakan pembaruan model pembayaran. Model pembayaran ini dikembangkan guna mendukung cash management yang baik dengan menyajikan informasi penerima serta pembayaran real time dengan memanfaatkan kemajuan teknologi.

Manfaat *E-billing* sendiri ada 5, yaitu :

- a. Mempermudah proses pembayaran dan menyederhanakan proses pengisian data
- b. Mengurangi kesalahan terhadap manusia dalam merekam data pembayaran.
- c. Memudahkan serta fleksibilitas cara membayar melalui beberapa alternatif saluran pembayaran.
- d. Memberikan akses kepada wajib bayar untuk memonitoring status pembayaran.
- e. Memberikan keleluasaan kepada wajib bayar dalam merekam data bayar secara mandiri.

2.6 MySQL

MySQL ialah *database server* yang banyak digunakan oleh orang. MySQL menggunakan *SQL* sebagai bahasa dasar untuk bisa mengakses *database*. MySQL merupakan *software database* yang bersifat *open source* serta server basis data yang memiliki kinerja cepat, *reliable* dan mudah digunakan.

2.7 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat sebuah website dan berfungsi untuk pengolahan data dari *user* (Andi :2007), Data dari *user* tersebut, akan disimpan di *database website* serta dapat kembali ditampilkan kembali apabila diakses.

2.8 Contingency Plan

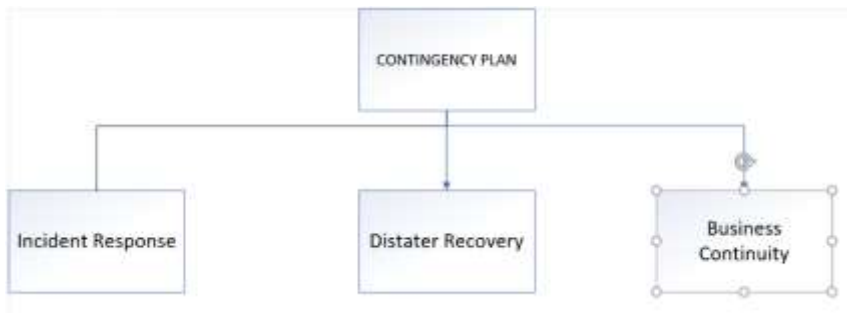
The National Institute of Standards and Technology menyatakan bahwa, "perencanaan kemungkinan IT mengacu pada strategi yang terkoordinasi yang melibatkan rencana, prosedur, dan langkah-langkah teknis yang memungkinkan pemulihan sistem TI, operasi, dan data setelah suatu gangguan" (Swanson et al., 2002). Selain itu, kontingensi juga harus dipersiapkan dan menjadi bagian dari organisasi yang lebih besar.

Sauter dan Carafano juga memberikan definisi ini untuk perencanaan kontingensi organisasi: "Sebuah rencana kontingensi adalah pernyataan komprehensif tindakan yang akan diambil sebelum,

selama, dan setelah bencana. Sebuah proses perencanaan yang sukses harus mencapai tiga sasaran: (1) menciptakan kesadaran dari potensi bencana, (2) menentukan tindakan dan kegiatan yang akan meminimalkan gangguan fungsi kritis, dan (3) mengembangkan kemampuan untuk membangun kembali operasi bisnis "(Sauter & Carafano 2005). Suatu kemungkinan harus diperhatikan dengan baik, ditulis dan diuji dalam sebuah *contingency plan*, yang akan dimasukkan kedalam rancangan darurat secara menyeluruh dan membantu perusahaan menahan dan memulihkan pasca bencana.

Jadi *contingency plan* adalah suatu perencanaan untuk kondisi yang tidak terduga. Perusahaan harus bersiap-siap untuk menghadapi, mendeteksi, bereaksi serta pemulihan dari pasca kondisi tersebut. Kondisi tersebut mengancam keamanan asset sebuah perusahaan. Tujuan dari *contingency plan* adalah melakukan perbaikan menuju mode operasi normal dengan biaya minimum serta gangguan gangguan aktivitas normal bisnis setelah terjadi bencana yang tak terduga.

2.8.1 Elemen *Contingency Plan*



Gambar 1-2.1 *Contingency Plan Hierarchies*

a. *Incident Response Planning* (IRP)

Respon yang akan diambil ketika terjadi insiden dengan segera.

b. *Disaster Recovery Planning* (DRP)

Melakukan perbaikan dilokasi utama setelah terjadi bencana.

c. *Business Continuity Planning* (BCP)

Merencanakan kelanjutan bisnis setelah dilakukannya perbaikan lokasi/situs.

2.9 Metode UCD(*User Centered Design*)

User centered design (UCD) disebut *human centered design*. *Human centered design* merupakan suatu metode yang dimanfaatkan untuk pengembangan sistem secara interaktif yang bertujuan untuk membuat sebuah perangkat lunak atau sistem (Ibnu Surya, et al: 2017). Definisi lainnya yaitu "*User Centered Design* (UCD) merupakan proses desain *interface* (antarmuka) pengguna yang berfokus pada tujuan kegunaan, karakteristik pengguna, lingkungan, tugas, dan alur kerja dalam desain antarmuka. UCD memiliki serangkaian metode yang terdefinisi dengan baik untuk analisis, desain, dan evaluasi perangkat keras, perangkat lunak, dan interface web utama. Proses UCD adalah proses berulang, di mana langkah-langkah desain dan evaluasi dibangun dari tahap pertama proyek, melalui implementasi." (Shawn Lawton Henry dan Mary Martinson, *Accessibility in User-Centered Design*). Beberapa prinsip yang harus diketahui mengenai metode UCD ini adalah :

a. Berfokus kepada yang menggunakan

- b. Perancangan integrasi
- c. Dari tahap awal berlanjut tahap pengujian
- d. Perancangan interaktif

2.9.1 Bagan dari Metode *User Centered Design* (UCD)



Gambar 2-2.2 Proses *User Centered Design*

a. *Specify the context of use*

Melakukan identifikasi tentang *user* yang menggunakan sistem ini seperti menjelaskan dalam kondisi apa mereka menggunakan sistem.

b. *Specify User and Organizational Requirements*

Melakukan identifikasi terhadap kebutuhan perusahaan.

c. *Design solutions*

Membuat design dengan data yang sudah teridentifikasi sebagai solusi.

d. Evaluate Design

Melakukan evaluasi terhadap sistem yang sudah dibuat.

2.10 Kuesioner

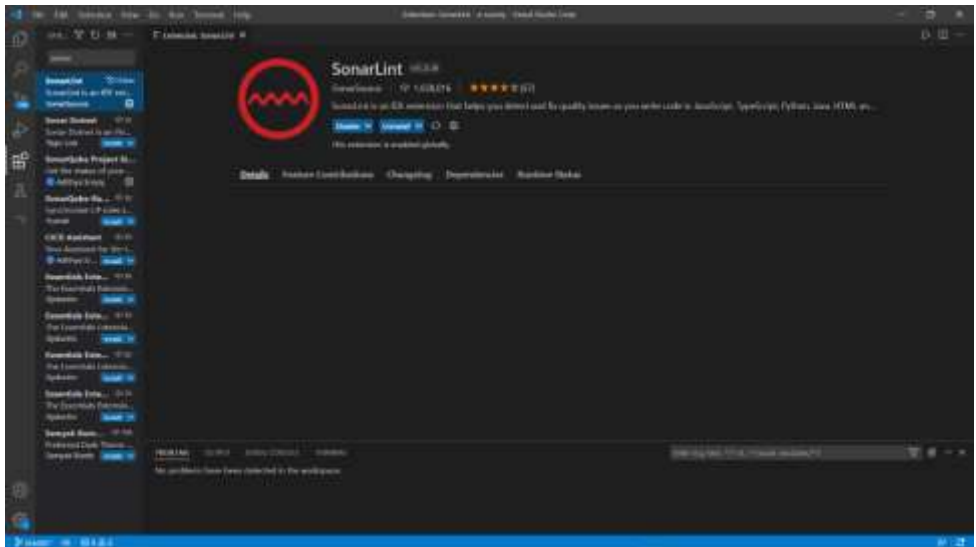
Angket atau kuesioner merupakan metode pengumpulan data yang dilakukan dengan cara memberi seperangkat pernyataan atau pertanyaan tertulis kepada responden untuk diberikan respon sesuai dengan permintaan pengguna. Kuesioner dalam penelitian ini digunakan untuk penentuan apakah dibutuhkan contingency plan terhadap sistem *billing* yang terdapat di terminal petikemas X. Kuesioner ini terdiri dari 10 pertanyaan yang tepat untuk mempertanyakan kelayakan *contingency plan*.

2.11 Sonarlint

Sonarlint merupakan tools ekstensi IDE yang dapat membantu mendeteksi serta memperbaiki masalah kualitas saat menulis sebuah kode program. *Sonarlint* dapat menemukan kekurangan pada kode program tersebut sehingga kode program tersebut diperbaiki.

Analisis Contingency Plan Sistem Billing Impor Menggunakan Metode User Centered Design

Sonarlint dapat digunakan di *Visual Studio Code* dengan menambahkan dimenu extension. Berikut merupakan gambar dari *sonarlint* yang berada di extension *VS.Code* :



Gambar 3-2.3 *Sonarlint*

Setelah terinstall di *VS. Code*, *sonarlint* akan memberitahu tentang kesalahan yang terjadi di dalam *codingan*. Berikut merupakan contoh ketika terjadi kesalahan terhadap *codingan*.



Gambar 4-2.4 Contoh kesalahan terdeteksi

Sonarlint juga memberitahu apakah codingan tersebut sesuai atau diganti sesuai dengan rekomendasi dari *sonarlint*



Gambar 5-2.5 Contoh Tidak Terjadi Kesalahan

tersebut. Namun ketika tidak terdeteksi kesalahan maka tampilan dari problemnya tidak akan muncul seperti gambar di bawah ini.

BAB III

Metode Penelitian

Pembahasan selanjutnya adalah tentang metode untuk pengembangan sistem *survey* ini karena diperlukan kaidah-kaidah yang berlaku agar penelitian dan hasilnya dinyatakan valid. Secara umum kita memerlukan metode penelitian untuk mendapatkan data secara ilmiah untuk tujuan dan kegunaan tertentu. Dalam penelitian ini, metode yang digunakan adalah Metode *User Centered Design* supaya penyelesaian penelitian berjalan secara sistematis dan metode ini memungkinkan adanya respon dari user terkait kekurangan dari sistem. Untuk rangkaian tahapan penelitian, digunakan metodologi penelitian SDLC (*System Development Life Cycle*).



Gambar 6-3.1 Diagram Alur Metodologi Penelitian

Alur metodologi penelitian ini memberikan gambaran tahapan penelitian di sebuah terminal petikemas X yang berada di Jakarta Utara.

3.1 Tahapan Diagram Alur Metodologi Penelitian

3.1.1 Analisis Kebutuhan

Kebutuhan yang perlu dipenuhi ialah pengumpulan data terkait dibutuhkan atau tidaknya metode UCD dengan *flowmap* dari sistem

yang sedang berjalan. Pengumpulan data dilakukan dengan cara berikut :

1. Observasi

Observasi dilakukan untuk mengamati serta mencatat sebuah permasalahan pada objek penelitian. Dalam penelitian ini, peneliti melakukan observasi pada:

Waktu : 24 Januari 2022

Tempat : Terminal Petikemas X Jakarta

Alamat : Digul No.1, RW.1, Koja, Kec. Koja, Kota Jakarta Utara, Daerah Khusus Ibukota Jakarta 14220.

Dalam hal ini, data yang berhasil dikumpulkan menyimpulkan diperlukannya proses *contingency plan* ketika terjadi insiden yang tidak terduga yang terjadi di *server* utama serta infrastruktur apa yang harus digunakan ketika *contingency plan* tersebut terlaksana.

2. Wawancara

Wawancara merupakan proses pengumpulan data yang berasal dari tanya jawab secara satu arah. Wawancara dilakukan kepada Koordinator *Programmer Analyst* dari terminal petikemas X yang berada di Jakarta. Wawancara dilakukan pada tanggal 24 Januari 2022 melalui tatap muka. Dalam wawancara ini, peneliti mendapatkan sebuah data berupa dokument yang dibutuhkan, perangkat komputer yang dibutuhkan untuk sistem dapat berjalan dengan baik. Berikut peneliti sudah mengelompokan data yang didapat.

Table 2-3.1 Tabel hasil wawancara

Dokumen yang dibutuhkan oleh Proses Impor	Dokumen SPPB (Surat Persetujuan Pengeluaran Barang)
	Dokumen DO (<i>Delivery Order</i>)
Perangkat Komputer penunjang sistem	Operasi Sistem menggunakan windows 7 dan windows 10
	Menggunakan RAM 4 GB
Kebutuhan Sistem	Menggunakan bahasa pemrograman PHP
	Menggunakan webserver berupa <i>tomcat</i> ataupun <i>Apache</i>
	Menggunakan Database Oracle
Permasalahan	membutuhkan sebuah <i>back up</i> sistem ketika terjadi insiden tidak terduga (<i>contingency Plan</i>)
Solusi	Contingency plan dilakukan dengan metode <i>local service</i>
	<i>Contingency plan</i> digunakan ketika terjadinya <i>offline server</i> utama

3.1.2 Desain Sistem

Dalam buku penelitian ini, desain sistem tidak hanya menjelaskan tentang *interface* saja, tetapi desain dalam metode ini. Berikut ini merupakan desain sistem yang dimaksud, yaitu :

1. Analisis Sistem Berjalan

Analisis sistem berjalan bertujuan untuk melakukan analisis sistem tentang kuisioner yang berjalan hingga saat ini.

2. Analisis Sistem Yang Dibangun

Analisis sistem yang akan dibangun bertujuan untuk membangun sebuah sistem survey mengenai dibutuhkannya atau tidak contingency plan di perusahaan terminal petikemas x.

3. Perancangan *Unified Modelling Language*

Analisis UML merupakan tahap untuk memberikan bahasa permodelan visual atau gambar untuk para *user* pemrograman yang berbagai macam serta proses umum rekayasa. UML (*Unified Modelling Language*) terdiri dari : *use case*, *Activity Diagram*, *Sequence Diagram*,

4. Analisis Metode *User Centered Design*

Analisis metode *User Centered Design* dilakukan untuk mengetahui apakah sistem *contingency plan* dibutuhkan untuk perusahaan atau tidak .

3.1.3 Implementasi

Proses berikutnya ialah melakukan pengkajian terhadap metode serta mengevaluasi metode *User Centered Design*. Dengan demikian peneliti mengetahui bisa atau tidaknya metode UCD diimplementasikan terhadap *contingency plan* sistem billing.

1. Implementasi Terhadap Metode *User Centered Design*

Implementasi Metode *User Centered Design* akan digunakan pada sistem e-kuesioner yang akan dibangun.

2. Implementasi Sistem dibangun

Sistem dibangun akan diimplementasikan ketika terjadi insiden yang tidak diinginkan oleh perusahaan sehingga sistem utama dapat digantikan oleh sistem *contingency plan*

3.1.4 Integrasi dan Pengujian

Proses selanjutnya adalah integrasi dan pengujian terhadap metode User Centered Design untuk mengetahui apakah sistem akan membantu dalam menentukan pentingnya sistem *contingency plan*.

1. Integrasi Metode *User Centered Design*

Metode UCD akan diintegrasikan setelah terbentuk *interface* dari sistem tersebut

2. Pengujian terhadap Metode *User Centered Design*

Pengujian terhadap Metode UCD akan dilakukan menggunakan *usability testing* yaitu *testing* berupa kuisisioner dari *user* terkait sistem *contingency plan*.

3.1.5 Pemeliharaan

Proses selanjutnya merupakan proses pemeliharaan terhadap sistem yang sudah dibangun supaya sistem dapat terpelihara baik serta dapat dilakukan pengembangan berikutnya.

3.2 Perumusan Metode *User Centered Design*

Peneliti menggunakan *usability testing* dalam metode *user centered design*. *Usability testing* diterapkan pada sistem e-kuesioner yang akan dibangun. Untuk para pengguna akan dibuatkan e-kuisiioner yang harus diisi supaya diketahui apakah *contingency plan* dibutuhkan dalam perusahaan atau tidak.

Usability Testing memiliki 10 pertanyaan yang hasilnya akan dikalkulasikan dengan menggunakan rumus pencarian rata-rata. Pertanyaan tersebut adalah

Table 3-3.2 Pertanyaan Kuisiioner *Usability Testing* sistem

No	Pertanyaan
1	Saya berpikir akan menggunakan sistem ini lagi
2	Saya merasa sistem Selalu terjadi gangguan
3	Saya merasa sistem ini mudah digunakan
4	Saya sistem sering tidak bisa digunakan
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6	Saya merasa jaringan internet perusahaan selalu bermasalah
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8	Saya merasa sistem selalu <i>not responding</i>

9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
10	Saya merasa bingung ketika sistem sedang terjadi gangguan

Setelah membuat kuisioner dengan pertanyaan tersebut, *usability testing* mempunyai 5 pilihan jawaban, yaitu sangat tidak setuju, tidak setuju, ragu-ragu, setuju, sangat setuju. Masing-masing jawaban memiliki skor dari 1 sampai 5. Berikut merupakan pilihan jawaban dari *usability testing* .

Table 4-3.3 Pilihan jawaban *Usability Testing*

Jawaban	skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju	2
Ragu-ragu	3
Setuju	4
Sangat Setuju	5

3.2.1 Perumusan *Usability Testing*

Setelah data dari responden terkumpul, proses selanjutnya adalah perhitungan . Dalam *usability testing* ada aturan-aturan dalam perhitungan skor. Berikut aturan-aturan dalam perhitungan skor pada kuisisioner :

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapatkan dari skor pengguna akan dikurangi 1.
2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
3. Skor *usability testing* didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

Aturan perhitungan skor tersebut berlaku untuk 1 responden. Perhitungan selanjutnya, skor *usability testing* dari masing-masing responden akan dicari rata-rata skor yang didapat dengan menjumlahkan seluruh hasil dari skor responden dan dibagi dengan banyak responden. Berikut rumus mencari rata-rata skor :

$$\bar{x} = \frac{\sum x}{n}$$

\bar{x}	=	skor rata-rata
$\sum x$	=	jumlah skor SUS
n	=	jumlah responden

Untuk mengetahui rata-rata skor masuk ke dalam kategori baik atau tidak, maka ditetapkan dengan rumus $100 \div 2 = 50$. Rumus tersebut merupakan penentuan kategori penilaian dari skor *usability testing*. Berikut tabel *range* kategori hasil akhir dari skor responden :

Table 5-3.4 Kategori Hasil Penilaian

50-100	Tidak dibutuhkan
0-49	Sangat Dibutuhkan

3.2.2 Contoh Perhitungan *Usability Testing*

Contoh dari *usability testing* menggunakan 4 responden dari kuisioner. Pertanyaan yang terdapat di kuisioner diubah menggunakan Q1 sampai Q10 dan angka merupakan skor dari jawaban atas kuisioner tersebut. Berikut contoh tabel dari responden :

Table 6-3.5 Contoh Responden *usability testing*

No	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Responden 1	5	2	5	3	5	2	5	3	5	3
2	Responden 2	4	2	5	2	4	3	4	3	4	3
3	Responden 3	4	3	4	1	5	3	4	3	4	3
4	Responden 4	5	3	5	2	4	2	5	1	5	2

Dari data diatas, untuk responden 1 sampai responden 5 terapkan aturan-aturan perhitungan skor *usability testing* yang sudah di jelaskan disub-bab perumusan *usability testing*. Setelah diterapkan aturan *usability testing*, maka hasilnya dijumlah serta hasil dari penjumlahan tersebut dikali dengan 2,5 sesuai aturan *usability testing*. Berikut hasil setelah diterapkan aturan-aturan *usability testing* :

Table 7-3.6 Tabel setelah dilakukan Aturan *Usability Testing*

No	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	JML	HASIL(*2,5)
1	Responden 1	4	3	4	2	4	3	4	2	4	2	32	80
2	Responden 2	3	3	4	3	3	2	3	2	3	2	28	70
3	Responden 3	3	2	3	4	4	2	3	2	3	2	28	70
4	Responden 4	4	2	4	3	3	3	4	4	4	3	34	85
Jumlah Hasil :													305
Rata-Rata Hasil Responden													76,25

Setelah didapat hasil dari aturan-aturan *usability testing*, proses perhitungan selanjutnya adalah mencari rata-rata dari hasil skor yang didapat menggunakan rumus :

$$\bar{x} = \frac{\sum x}{n}$$

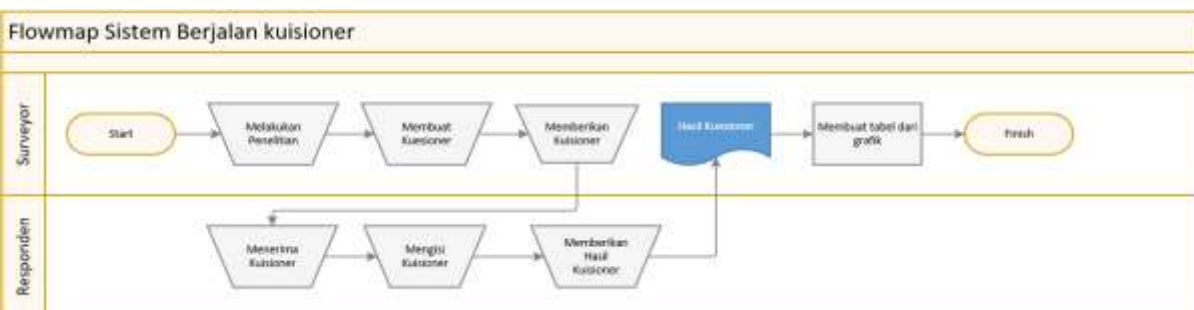
Jumlah dari hasil skor responden kuisioner dibagi dengan jumlah responden yang tersedia, maka hasil nya adalah **76,25** dan dibulatkan menjadi **76**. Dan hasil tersebut termasuk kedalam kategori **TIDAK MEMBUTUHKAN**.

BAB IV

PEMBAHASAN HASIL ANALISIS

Dalam buku ini, bab selanjutnya membahas tentang hasil dari analisis yang sudah peneliti lakukan. Bab ini akan mempresentasikan berupa *flowmap* sistem yang sedang berjalan, *flowmap* sistem yang akan dibangun, perancangan UML, perancangan *Database* serta perancangan interface dalam sistem yang akan dibangun. Pada sub-bab selanjutnya akan membahas tentang *flowmap* sistem yang sedang berjalan.

4.1 *Flowchart* Sistem Yang Sedang Berjalan



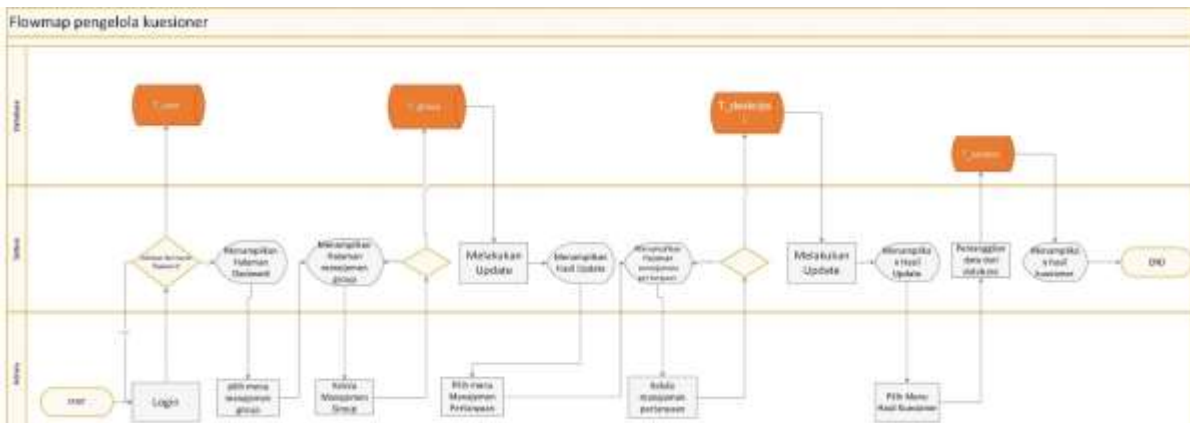
Gambar 7-4.1 *Flowchart* yang berjalan

Keterangan dari *flowchart* ini adalah :

1. Surveyor melakukan penelitian dan menentukan target untuk mengisi kuesioner tersebut sesuai dengan penelitiannya..
2. Setelah menentukan target yang digunakan untuk pengisian kuisisioner tersebut, maka surveyor akan membuat kuisisioner tentang penelitian surveyor tersebut.
3. Setelah dibuat kuesioner, maka surveyor menyerahkan kepada target dari pengisian tersebut.
4. Responden melakukan pengisian kuesioner tersebut sesuai dengan yang di berikan surveyor.
5. Setelah responden selesai mengisi kuesioner, Responden memberikan kembali hasil Kuesioner kepada surveyor.
6. Setelah hasil di terima, surveyor akan membuat tabel dari hasil responden tersebut dan dibuatkan grafik yang menjadi hasil dari kuesioner tersebut.

4.2 *Flowchart* Sistem Yang Akan Dibangun

4.2.1 Admin

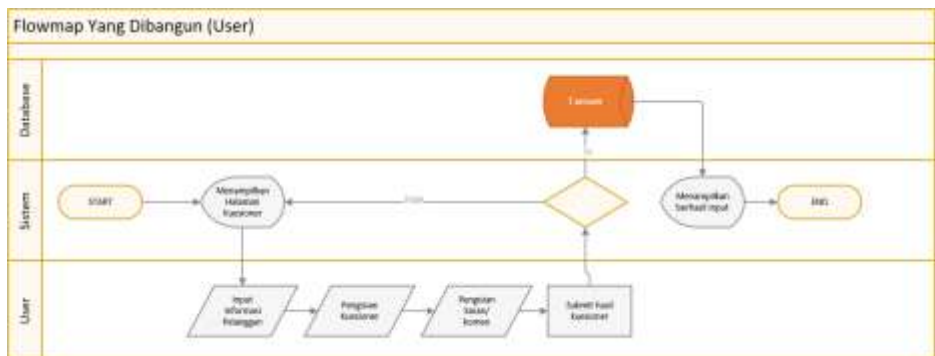


Keterangan dari *Flowchart* diatas adalah :

1. Admin Terminal Petikemas X terlebih dahulu masuk kedalam sistem.
2. Admin Terminal Petikemas X melakukan login terlebih dahulu. Kemudian sistem akan mevalidasi *username* dan *password* sudah sesuai dengan database atau tidak, jika sesuai maka akan dilanjutkan dengan halaman *dashboard* admin. Jika tidak maka kembali ke halaman utama.
3. Admin melakukan pemilihan terhadap menu *dashboard* admin yang di tampilkan. Selanjutnya admin akan menuju ke *menu* manajemen kategori, yang kemudian sistem akan menampilkan halaman dari manajemen kategori.
4. Admin akan meneglola kategori pertanyaan yang berfungsi untuk mengkategorikan setiap pertanyaan untuk memudahkan perusahaan dalam menilai pertanyaan tersebut sesuai kategori.
5. Setelah sukses membuat kategori, sistem akan melakukan pengecekan apakah masih ada yang kurang atau tidak, setelah aman, sistem akan menyimpan di dalam *database* kemudian akan menampilkan data dari kategori yang sudah dibuat oleh admin.
6. Kemudian admin menuju ke dalam menu manajemen pertanyaan. Kemudian sistem akan menampilkan halaman manajemen pertanyaan.

7. Kemudian di dalam *menu* manajemen pertanyaan, admin melakukan pengelolaan pertanyaan bisa berupa membuat, menghapus atau mengedit data pertanyaan tersebut. Sistem akan melakukan pengecekan apakah masih ada yang kurang atau tidak, setelah aman, sistem akan memproses dan akan di *update* di *database*.
8. Selanjutnya admin akan melihat hasil kuesioner tersebut di dalam *menu* hasil kuesioner. Hasil tersebut sudah di terapkan aturan aturan *usability testing*.

4.2.2 User



Gambar 9-4.3 Flowmap pengisian kuesioner

Keterangan dari *flowmap* ini adalah :

1. sistem akan menampilkan halaman kuesioner
2. user melakukan *input* informasi pelanggan berupa nama pelanggan, alamat, telpon.
3. Setelah *user* melakukan *input* informasi pelanggan, pengguna melanjutkan pengisian terhadap kuesioner yang terdapat di sistem tersebut

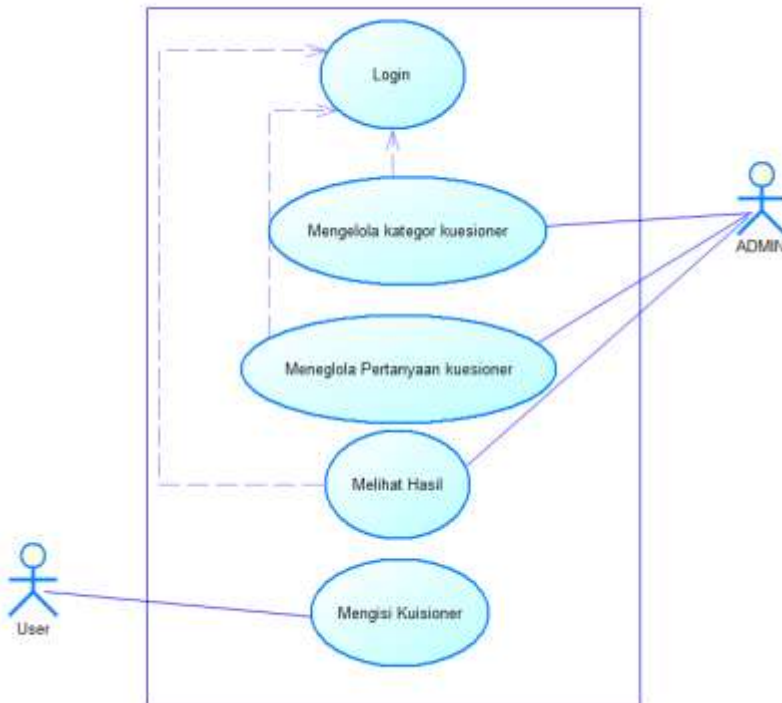
4. Setelah pengisian kuesioner selesai, pengguna diharapkan memberikan saran ataupun komentar tentang sistem tersebut.
5. Setelah semua telah terisi, pengguna menekan tombol *submit* yang tertera dibawah kolom saran.
6. Sistem akan melakukan pengecekan terhadap pengisian tersebut apakah masih ada yang kosong atau tidak. Jika sudah terisi semua, sistem akan menyimpan data hasil kuesioner tersebut kedalam *database*.
7. Setelah berhasil terinput, sistem akan menampilkan halaman terima kasih sudah melakukan pengisian terhadap kuesioner tersebut.

4.3 Perancangan

Pada Proses Perancangan akan menjelaskan tentang beberapa gambaran dalam menganalisis serta membangun sebuah sistem. Gambaran tersebut berupa perancangan *Usecase*, perancangan *class diagram*, *sequence diagram*, *database*, *User Interface* serta perangkat lunak dan keras yang di gunakan.

4.3.1 *Usecase Diagram*

Diagram *usecase* berikut akan menjelaskan tentang aktor dan kegiatan yang berada didalam sistem tersebut.



Gambar 10-4.4 *Usecase Diagram*

4.3.1.1 Definisi Aktor

Table 8-4.1 Definisi Aktor

No	Aktor	Keterangan
1	Admin	Dapat melakukan pengelolaan terhadap kategori, pertanyaan serta dapat melihat hasil kuesioner .

2	User	User hanya dapat mengisi kuesioner yang telah di buat oleh admin.
---	------	---

4.3.1.2 Definisi *Usecase*

Table 9-4.2 Definisi *Usecase*

Definisi Usecase	Keterangan
<i>Login</i>	Aktifitas admin yang dapat mengakses halamana admin
Mengelola Kategori Pertanyaan	Aktifitas admin untuk mengelola kategori dari pertanyaan
Mengelola Pertanyaan	Aktifitas admin mengelola pertanyaan-pertanyaan yang akan di berikan oleh user.
Melihat Hasil	Aktifitas admin yang dapat mengakses dan melihat hasil dari kuesioner tersebut di halaman hasil
Mengisi Kuesioner	Aktifitas user yang untuk mengisi kuesioner yang telah di sediakan oleh sistem

4.3.1.3 Skenario *Usecase*

Table 10-4.3 Skenario *Usecase Login*

Identifikasi	
Nama	Login
Tujuan	Untuk Mengakses semua Fungsi
Deskripsi	Memberikan akses pengguna terhadap sistem dengan melakukan validasi terhadap <i>username</i> , <i>password</i> yang di masukan oleh pengguna
Aktor	Admin
Skenario	
Kondisi Awal	Pilih <i>button login</i> untuk masuk ke form login
Reaksi Aktor	Reaksi Sistem
memasukkan Username dan Password	Form <i>login</i> akan menampilkan <i>textbox username, password</i> . Password yang ditampilkan berupa kode sehingga keamanan terjaga

Admin melakukan persetujuan tentang username dan password dengan menekan tombol masuk	Aplikasi akan memvalidasi terhadap <i>username</i> dan <i>password</i> dari pengguna dengan pengecekan <i>database</i>
Kondisi Akhir	Jika <i>Username</i> dan <i>password</i> yang di masukam oleh pengguna maka aktor akan masuk kedalam halaman utama admin, dan dapat menggunakan sistem dengan hak aksesnya.

Table 11-4.4 skenario Kelola kategori pertanyaan

Identifikasi	
Nama	Mengelola Kategori Kuesioner
Tujuan	mengkategorikan setiap pertanyaan
Deskripsi	
Aktor	Admin
Skenario	

Kondisi Awal	Admin harus <i>login</i> terlebih dahulu, jika valid maka masuk ke menu utama admin, jika tidak maka akan muncul <i>username</i> atau <i>password</i> salah
Reaksi Aktor	Reaksi Sistem
Memulai Aplikasi	Menampilkan halaman utama admin
1. Admin Memilih menu yang akan dilihat	sistem akan menampilkan menu yang diminta
2. Admin memilih aksi yang akan dilakukan, tambah kategori, edit kategori atau hapus kategori	sistem akan melakukan respon sesuai aksi dari pengguna
3. Admin melakukan konfirmasi atas aksi yang dipilih	sistem akan menampilkan hasil dari aksi yang dipilih
Kondisi Akhir	sistem akan menampilkan pembaruan terhadap menu tersebut sesuai dengan aksi yang di konfirmasi admin.

Table 12-4.5 skenario kelola pertanyaan

Identifikasi	
Nama	Mengelola Pertanyaan Kuesioner
Tujuan	memberikan pertanyaan kepada responden atau pengguna
Deskripsi	untuk menambahkan, mengubah serta menghapus pertanyaan untuk responden
Aktor	Admin
Skenario	
Kondisi Awal	Admin harus <i>login</i> terlebih dahulu, jika valid maka masuk ke menu utama admin, jika tidak maka akan muncul <i>username</i> atau <i>password</i> salah
Reaksi Aktor	Reaksi Sistem
Memulai Aplikasi	Menampilkan halaman utama admin
1. Admin Memilih menu yang akan dilihat	sistem akan menampilkan menu yang diminta
2. Admin memilih aksi yang akan dilakukan, tambah kategori, edit kategori atau hapus kategori	sistem akan melakukan respon sesuai aksi dari pengguna

3. Admin melakukan konfirmasi atas aksi yang dipilih	sistem akan menampilkan hasil dari aksi yang dipilih
Kondisi Akhir	sistem akan menampilkan perbaruan terhadap menu tersebut sesuai dengan aksi yang di konfirmasi admin.

Table 13-4.6 Skenario hasil kuesioner

Identifikasi	
Nama	Hasil Kuesioner
Tujuan	Melihat jawaban dari kuesioner pengguna
Deskripsi	untuk melihat hasil dari kuesioner pengguna dan perhitungan dari <i>usability testing</i>
Aktor	Admin
Skenario	
Kondisi Awal	Admin harus <i>login</i> terlebih dahulu, jika valid maka masuk ke menu utama admin, jika tidak maka akan muncul <i>username</i> atau <i>password</i> salah
Reaksi Aktor	Reaksi Sistem
Memulai Aplikasi	Menampilkan halaman utama admin

1. Admin Memilih menu yang akan dilihat	sistem akan menampilkan menu yang diminta
2. Admin akan melakukan aksi berupa pemilihan data keseluruhan atau data perkategori	sistem akan melakukan respon sesuai aksi dari pengguna
3. Admin melakukan konfirmasi atas aksi yang dipilih	sistem akan menampilkan hasil dari aksi yang dipilih
Kondisi Akhir	sistem akan menampilkan aksi yang dipilih berupa data keseluruhan atau data per kategori sesuai yang di pilih admin

Table 14-4.7 Skenario pengisian kuesioner

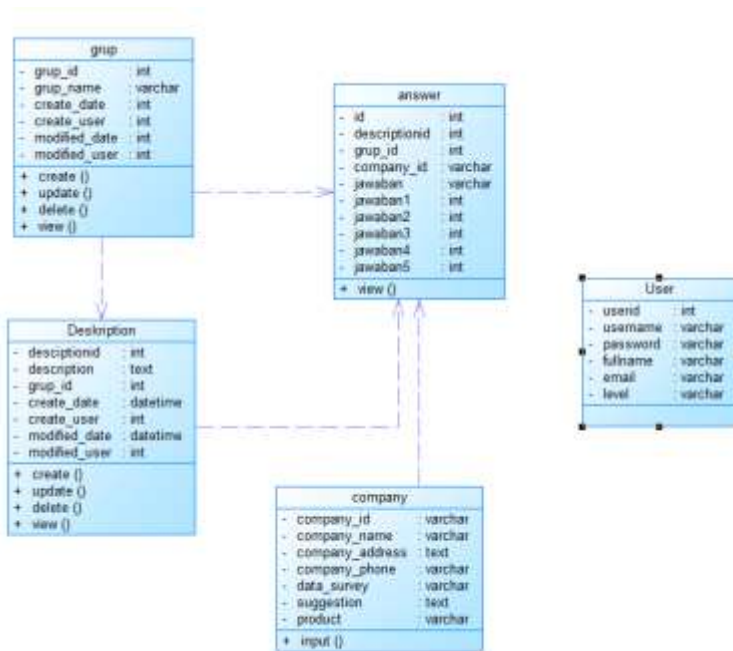
Identifikasi	
Nama	Mengisi Kuesioner
Tujuan	Untuk mengisi pertanyaan
Deskripsi	Melakukan pengisian terhadap pertanyaan kuesioner yang teradapt didalam sistem

Aktor	User
Skenario	
Kondisi Awal	telah tersedia form pengisian kuesioner di halaman utama
Reaksi Aktor	Reaksi Sistem
user memasukkan nama pelanggan, alamat, pilihan produk, serta nomer handphone	Halaman Utama akan menampilkan form pengisian informasi pelanggan berupa <i>textbox</i> nama pelanggan, alamat, serta nomer handphone. Sistem juga memberikan pilihan produk berupa transaksi impor maupun ekspor, serta tanggal sesuai dengan <i>realtime</i>
user melakukan pengisian terhadap pertanyaan yang diberikan oleh sistem dengan memilih dari option dari setiap pertanyaan .	Apliasi akan merespon pengisian yang dilakukan oleh <i>user</i>
user mengisi kolom komen/saran untuk perusahaan	aplikasi merespon pengisian yang dilakukan oleh <i>user</i> di kolom komen/saran

User melakukan validasi terkait pengisian kuesioner serta saran dengan menekan tombol submit	Aplikasi akan melakukan validasi terkait semua data sudah terisi dan tidak ada yang kosong sehingga sistem akan mengirimkan kedalam <i>database</i>
Kondisi Akhir	Jika pada akhir interaksi semua data telah terisi maka sistem akan menampilkan halaman ucapan terima kasih

4.3.2 *Class Diagram*

Fungsi dari *class diagram* ialah menggambarkan hubungan antar objek dan struktur yang terdapat didalam sistem. Struktur merupakan atribut-atribut dan *method-method* yang ada di setiap kelas. *Class Diagram* pada sistem *survey contingency plan* sistem *billing* adalah .



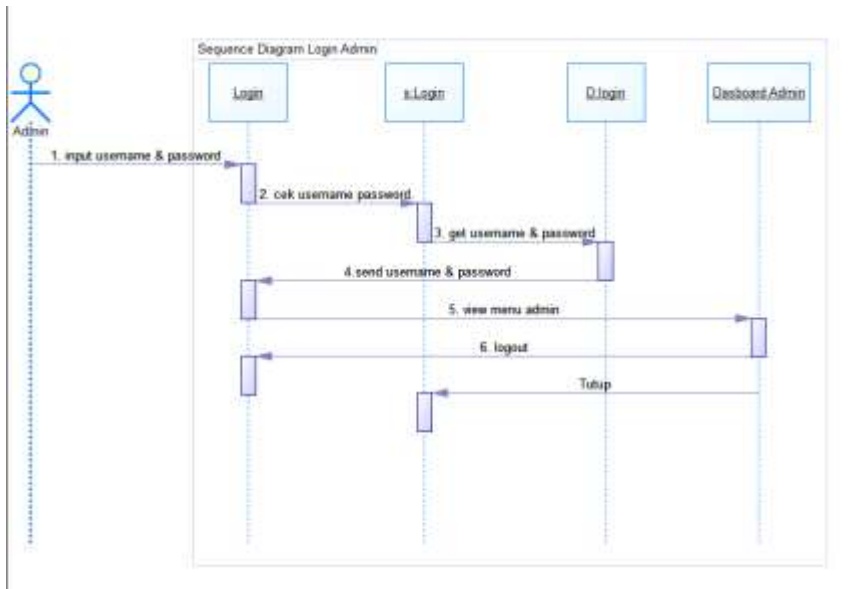
Gambar 11-4.5 Class Diagram sistem survey contingency plan sistem billing

4.3.3 Sequence Diagram

Sequence diagram ialah diagram yang menggambarkan dan menjelaskan objek-objek dalam sebuah sistem secara terperinci. Sistem akan melakukan sesuai apa yang objek perintahkan.

4.3.3.1 Sequence Diagram Admin bagian Login

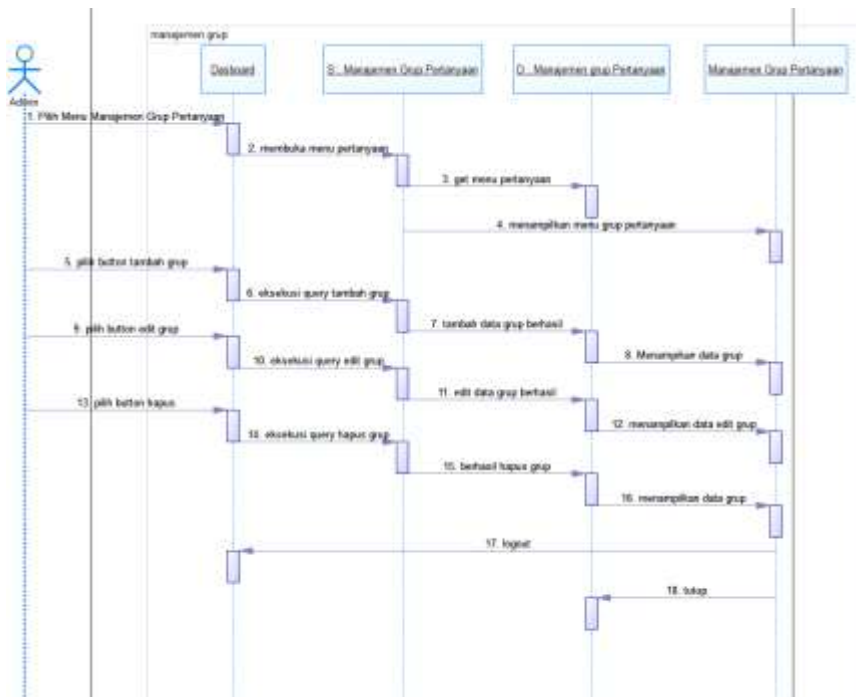
Berikut *sequence diagram login* yang menjelaskan tentang hubungan admin, halaman *login* dan *menu dashboard* admin



Gambar 12-4.6 Sequence Diagram Admin Login

Sequence diagram ini mendeskripsikan alur dari login. Admin akan menjalankan sistem dan menampilkan halaman utama berupa kuesioner . kemudian admin memilih halaman *login* dan akan tampil *form login*.kemudia aktor akan menginputkan *username* dan *password* kedalam sistem. Sistem akan melakukan pengecekan terhadap *username* dan *password* aktor di dalam *database*. Setelah dinyatakan valid maka sistem akan langsung menampilkan halaman *dasboard* dari admin.

4.3.3.2 Sequence Diagram Admin Manajemen Kategori Pertanyaan

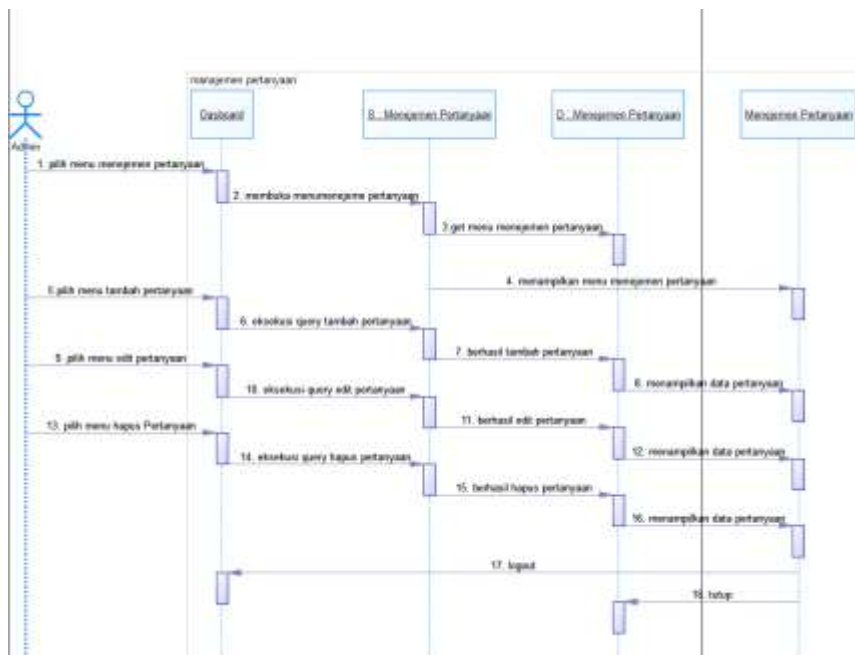


Gambar 13-4.7 Sequence Diagram Menejemen Kategori Pertanyaan

penjelasan sequence diagram ini ialah proses admin memanajemen kategori dari pertanyaan. Aktor terlebih dahulu untuk melakukan *login*. Selanjutnya sistem akan menampilkan halaman utama dari admin. Untuk aktor memulai kelola kategori pertanyaan, terlebih dahulu memilih *menu* manajemen kategory pertanyaan. Kemudian sistem akan melakukan konektivitas terhadap *database*. Setelah halaman kategori pertanyaan tampil, aktor dapat melakukan tambah data, edit data, serta hapus data dari kategori pertanyaan.

Jika semua sudah terisi, maka sistem akan melakukan pengecekan guna mengetahui apakah operasi tersebut berhasil atau gagal.

4.3.3.3 Sequence Diagram Admin Menejemen Pertanyaan

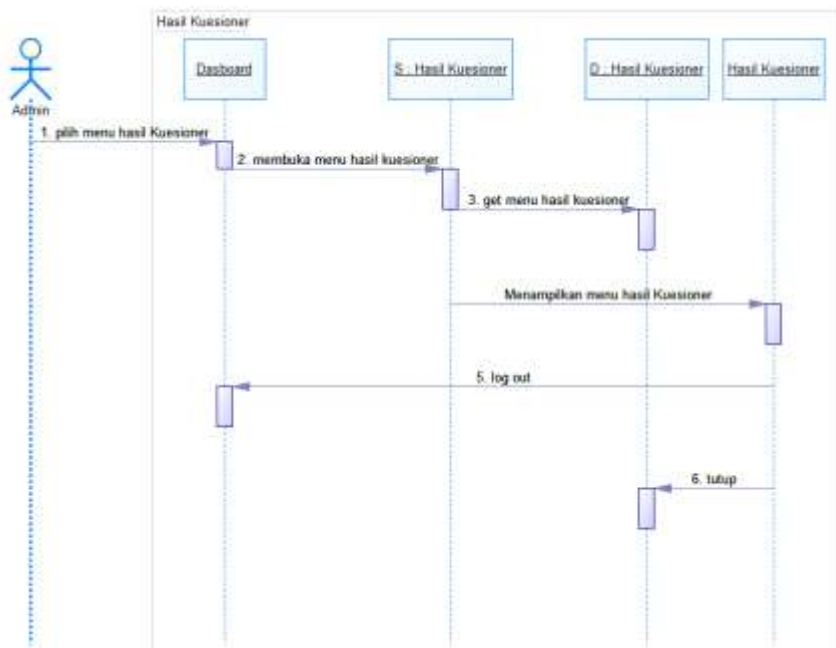


Gambar 14-4.8 Sequence Diagram Menejemen Pertanyaan

penjelasan *sequence diagram* ini ialah proses admin memanajemen pertanyaan. Aktor terlebih dahulu untuk melakukan *login*. Selanjutnya sistem akan menampilkan halaman utama dari admin. Untuk aktor memulai kelola pertanyaan, terlebih dahulu memilih *menu* manajemen pertanyaan. Kemudian sistem akan melakukan konektivitas terhadap *database*. Setelah halaman kategori pertanyaan tampil, aktor dapat melakukan tambah data, edit data, serta hapus data dari kategori pertanyaan. Jika semua sudah terisi, maka

sistem akan melakukan pengecekan guna mengetahui apakah operasi tersebut berhasil atau gagal.

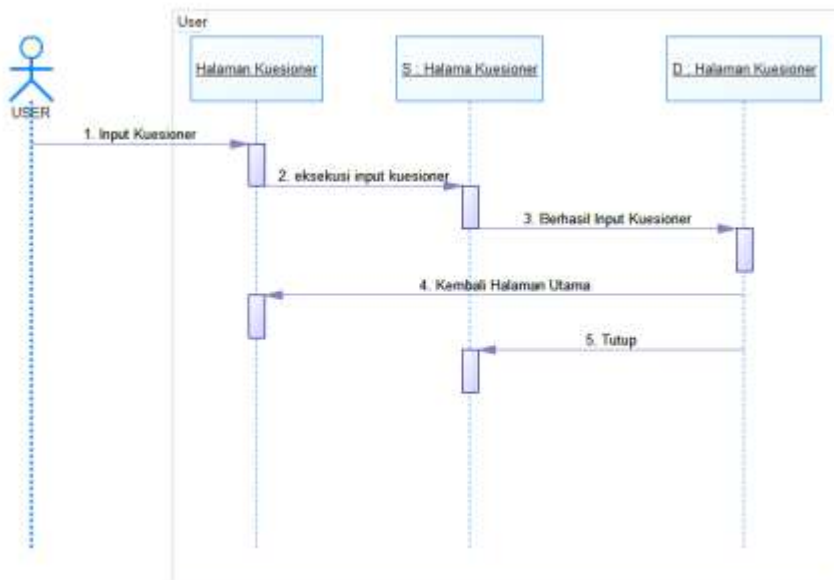
4.3.3.4 Sequence Diagram Admin Bagian Hasil Kuisisioner



Gambar 15-4.9 Sequence Hasil Kuisisioner

penjelasan *sequence diagram* ini ialah proses admin melihat hasil kuisisioner. Aktor terlebih dahulu untuk melakukan *login*. Selanjutnya sistem akan menampilkan halaman utama dari admin. Untuk aktor melihat hasil dari kuisisioner tersebut, terlebih dahulu memilih *menu* hasil kuisisioner. Kemudian sistem akan melakukan konektivitas terhadap *database*. Setelah halaman hasil kuisisioner tampil, aktor dapat melihat hasil kuisisioner yang telah diisi responden

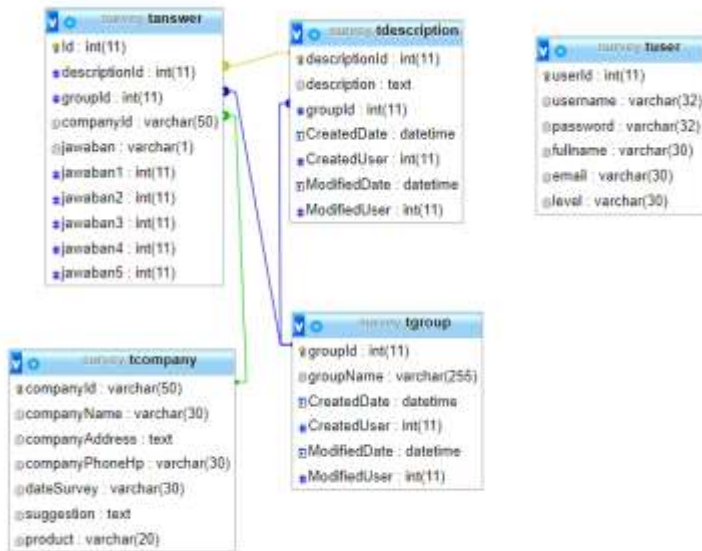
4.3.3.5 Sequence Diagram User



Gambar 16-4.10 Sequence Diagram User

Sequence diagram ini menjelaskan tentang user menggunakan sistem dalam mengisi kuesioner. Ketika *user* menggunakan sistem ini, maka langsung ditampilkan *menu* dari kuesioner tersebut. Ketika semua sudah diinputkan oleh *user*, maka sistem akan melakukan pengecekan terhadap setiap kolom dari kuesioner tersebut apakah ada yang tertinggal atau tidak. Jika sudah lengkap, sistem akan mengirimkan data tersebut kedalam *database*.

4.3.4 Perancangan Database



Gambar 17-4.11 Perancangan Database

Perancangan *database* bertujuan memberi gambaran tentang tabel dan *field* yang di gunakan dalam sistem yang akan di bangun. Pada perancangan ini, menggambarkan tabel yang saling berelasi terhadap tabel yang lain.

4.3.5 Perancangan *Interface*

Perancangan *interface* dilakukan untuk menterjemah bahasa sistem menjadi tampilan pada layar yang memudahkan pengguna dalam mengoperasikan sistem tersebut. Berikut adalah *interface* yang terdapat di sistem *e-survey contingency plan* sistem *billing*.

4.3.5.1 *Interface* Pengguna



Gambar 18-4.12 Halaman Utama User

Interface ini merupakan tampilan dari halaman utama dari pengguna. Pengguna tidak perlu melakukan *login* terlebih dahulu, namun harus mengisi informasi pelanggan. Informasi pelanggan terdiri dari nama pelanggan, alamat, produk, No Handphone serta tanggal. Namun, tanggal secara otomatis akan *update* secara realtime sehingga pengguna tidak perlu lagi melakukan pemilihan terhadap tanggal.

The screenshot shows a web browser window displaying a questionnaire titled "KUESIONER". The interface is divided into two main sections: "Pendahuluan" (Introduction) and "Isi" (Content). The "Pendahuluan" section contains a paragraph of text and a "Lanjut" (Next) button. The "Isi" section contains a table with multiple columns and rows, each with a radio button for selection. Below the table, there is a text area for "Komentar / Saran" (Comments / Suggestions) and a "Selesai" (Finish) button. The interface is clean and professional, with a dark header and a light background for the questionnaire content.

Gambar 19-4.13 Halaman Kuesioner

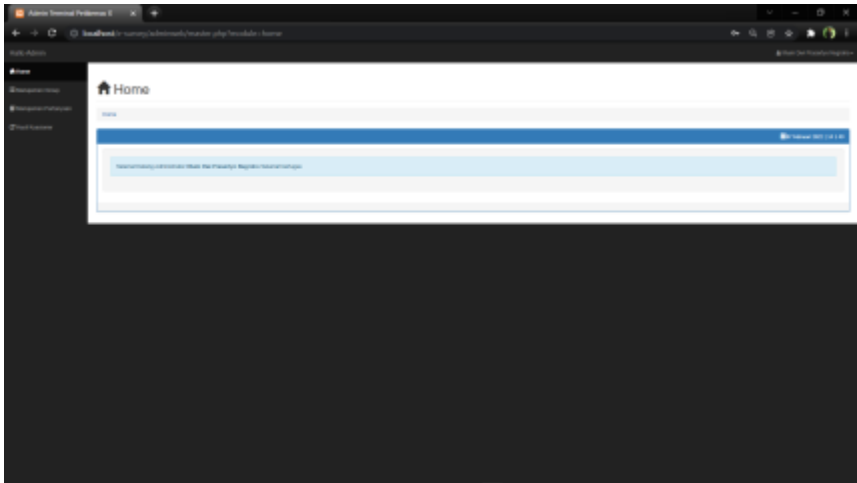
Selanjutnya pada perancangan ini, halaman kuesioner merupakan halaman pengguna dalam mengisi kuesioner yang telah tersedia dalam sistem tersebut. Cara pengguna dalam mengisi kuesioner tersebut dengan menandai lingkaran yang terdapat di kolom respon. Pengisian dilakukan dengan jujur, dikarenakan pengisian kuesioner tersebut akan menentukan dibutuhkannya *contingency plan* terhadap sistem *billing* tersebut atau tidak. Pengguna juga bisa memberikan komentar atau saran di kolom yang sudah disediakan. Kolom komen atau saran tersebut berguna untuk memperbaiki kekurangan dari sistem *billing* yang berjalan dan akan diimplementasikan di sistem *contingency plan* sehingga pengguna merasa nyaman menggunakan sistem tersebut,

4.3.5.2i Admin



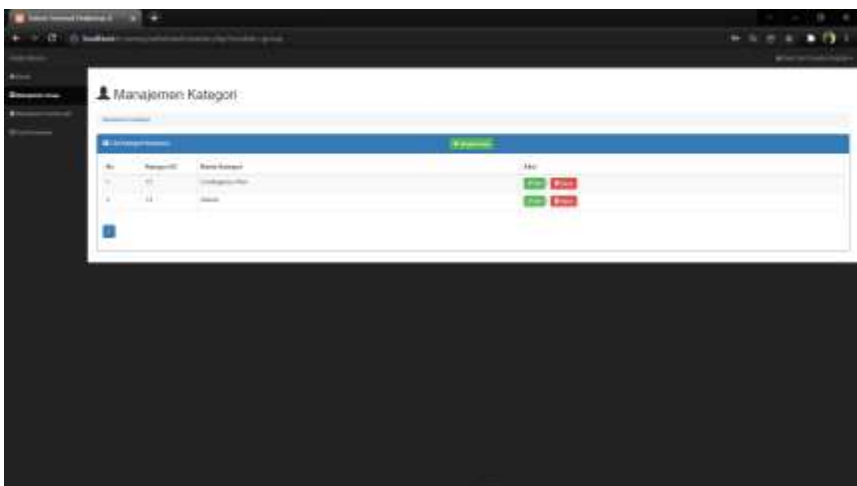
Gambar 20-4.14 Halaman *Login*

Halaman *login* diperuntukan untuk admin dari sistem ini. Admin akan melakukan *login* untuk dapat mengakses *menu* admin dan mengelola sistem ini. Admin dapat mengelola kategori pertanyaan, mengelola pertanyaan serta admin dapat mengetahui hasil dari kuesioner tersebut, baik secara keseluruhan atau secara per kategori.



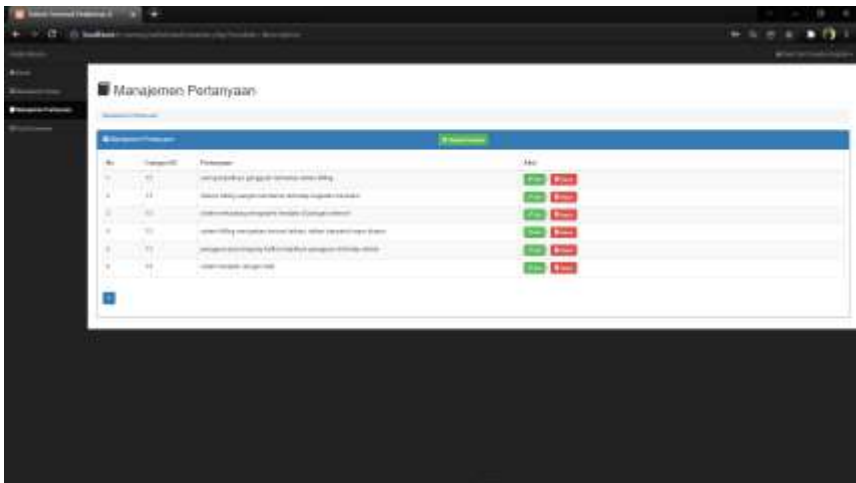
Gambar 21-4.15 *Dashboard Admin*

Setelah melakukan *login*, admin akan diarahkan menuju *dashboard* admin, kemudian admin akan melaksanakan tugas admin yaitu, mengelola kategori pertanyaan, mengelola data pertanyaan serta melihat hasil dari kuesioner tersebut.



Gambar 22-4.16 Halaman Manajemen Kategori

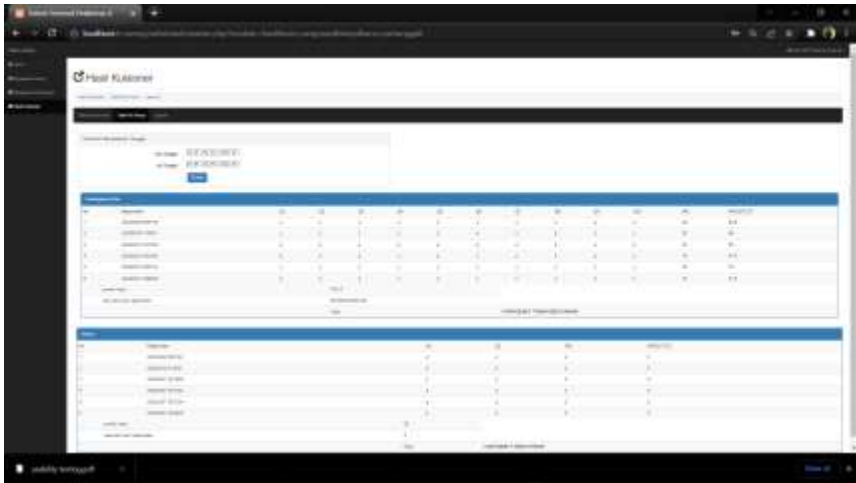
Halaman manajemen kategori pertanyaan, admin bertugas untuk mengkategorikan pertanyaan tersebut. Tujuan mengkategorikan pertanyaan ialah, untuk memudahkan dalam melihat hasil.



Gambar 23-4.17 Halaman Manajemen Pertanyaan

Interface ini merupakan menu untuk manajemen pertanyaan atau mengelola pertanyaan. Dalam menu ini, admin bisa menambahkan pertanyaan, mengedit pertanyaan serta menghapus pertanyaan. Pertanyaan yang ada di *menu* ini, akan tampil di halaman utama pengguna sebagai kuesioner yang harus diisi oleh para pengguna.

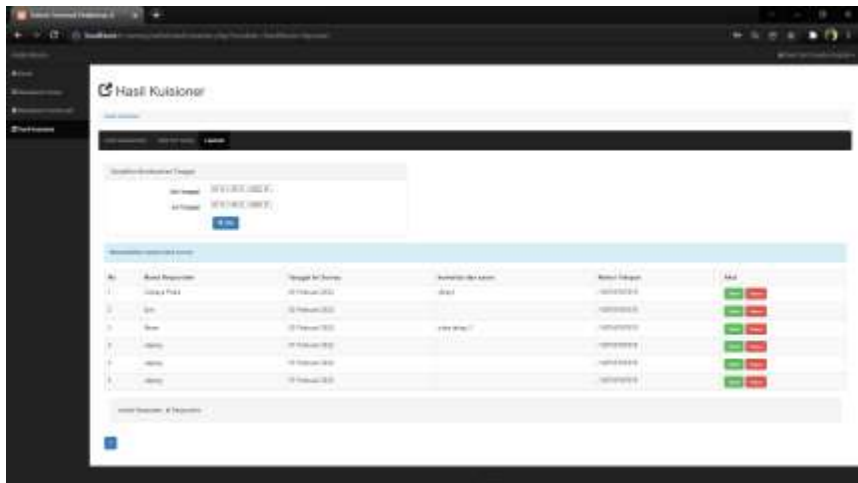
Pada menu ini, admin bisa melihat hasil dari kuesioner yang sudah dilakukan. Menu ini di bagi menjadi 3 bagian, yaitu : Data Keseluruhan, Data per kategori dan laporan. Bagian tersebut sama saja, yang membedakan adalah isi dari submenu tersebut. Untuk data keseluruhan adalah data yang di dapat dari kuesioner dari seluruh kategori. Jika hanya ada 1 kategori saja, maka bisa di pastikan hasilnya merupakan penentuan dibutuhkannya contingency plan atau tidak. Jika admin membuat 2 kategori, makan bisa ke submenu yaitu data per kategori.



Kategori	Subkategori	Pertanyaan	Jawab	Skor	Rata-rata
Kategori 1	Subkategori 1	Pertanyaan 1.1	Jawab 1.1	1	1.0
		Pertanyaan 1.2	Jawab 1.2	2	2.0
		Pertanyaan 1.3	Jawab 1.3	3	3.0
		Pertanyaan 1.4	Jawab 1.4	4	4.0
Kategori 2	Subkategori 2	Pertanyaan 2.1	Jawab 2.1	1	1.0
		Pertanyaan 2.2	Jawab 2.2	2	2.0
		Pertanyaan 2.3	Jawab 2.3	3	3.0
		Pertanyaan 2.4	Jawab 2.4	4	4.0
Kategori 3	Subkategori 3	Pertanyaan 3.1	Jawab 3.1	1	1.0
		Pertanyaan 3.2	Jawab 3.2	2	2.0
		Pertanyaan 3.3	Jawab 3.3	3	3.0
		Pertanyaan 3.4	Jawab 3.4	4	4.0

Gambar 25-4.19 Halama Hasil Per kategori

Submenu hasil per kategori merupakan submenu jika memiliki lebih dari 1 kategori, submenu ini digunakan untuk mempermudah melihat hasil dari kategori yang dijadikan kuesioner. Sehingga tidak melihat ke halaman pengguna untuk mengetahui pertanyaan yang berbeda kategori. Submenu pergroup ini bisa menampilkan data berdasarkan tanggal yang ingin di tampilkan.



Gambar 26-4.20 Halaman Laporan

Submenu laporan merupakan submenu untuk melihat detail dari para responden dan melihat saran apa yang diberikan terhadap sistem ini. Saran dari responden sangat dibutuhkan demi meningkatkan kualitas dari sistem perusahaan sehingga pengguna juga merasa nyaman menggunakan sistem tersebut.

4.3.6 Perancangan Perangkat Lunak dan Keras

Perancangan perangkat lunak dan keras dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis pengguna/*User*. Adapun kebutuhan perangkat lunak dan perangkat keras yang akan di buat adalah:

4.3.6.1 Kebutuhan Perangkat Keras(*Hardware*)

Kebutuhan *hardware* dalam pembuatan sistem ini sebagai berikut:

Table 15-4.8 *Hardware*

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Hardisk</i>	1 TB	Sebagai tempat untuk menyimpan data yang dibutuhkan tetapi pada sisi <i>client</i> tidak diharuskan memiliki ketersediaan <i>space</i> yang besar
2	<i>Memory RAM</i>	8 GB	Kecepatan <i>client</i> dalam mengakses <i>system</i> ini
3	<i>Processor</i>	<i>Intel® core™ i7-8750H CPU @2,20 Ghz</i>	Untuk per Halamanan komputer

4.3.6.2 Kebutuhan Perangkat Lunak (*software*)

Dalam perancangan ini, *software* yang digunakan dalam pembangunan sistem ini sebagai berikut :

Table 16-4.9 *Software*

No	Tools / Software	Fungsi	Keterangan
1.	<i>Windows</i>	Sistem Operasi	Penghubung antara aplikasi dan perangkat keras
2.	<i>Xampp</i>	<i>Server</i>	Penerjemah bahasa yang

		Basis Data	ditulis dengan bahasa pemrograman PHP
4.	<i>VS.Code</i>	<i>Text editor</i>	Untuk melakukan coding sistem
5.	<i>Google Chrome,</i>	<i>Browser</i>	Mencari berita di internet

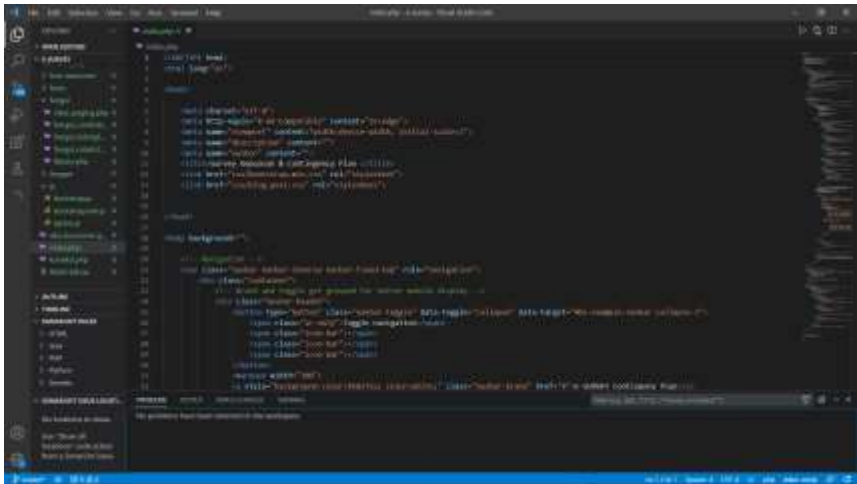
4.4 Pengujian

Pada tahap pengujian, peneliti melakukan 2 pengujian yaitu *white box* dan *usability testing*. Pengujian *White box* bertujuan untuk mengetahui apakah terjadi kesalahan terhadap codingan yang sudah di buat. Sedangkan untuk *usability testing* bertujuan menentukan penilaian untuk menentukan pentingnya *contingency plan* atau tidak

4.4.1 *White Box*

Pada pengujian *white box* ini, peneliti menggunakan *sonarlint* untuk menguji codingan yang telah dibuat dalam membangun *sistem e-kuesioner*. Berikut pengujian yang dilakukan disetiap codingan yang telah dibuat :

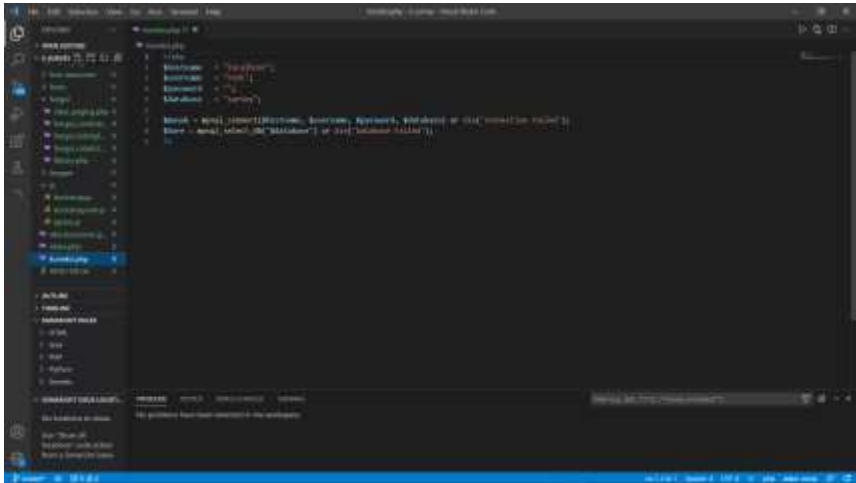
4.4.1.1 Pengujian Halaman Utama



Gambar 27-4.21 Pengujian Halaman Utama

Pada pengujian ini, *sonarlint* tidak mendeteksi adanya kesalahan yang terjadi pada codingan halaman utama yaitu tempat pengisian informasi pelanggan, tempat pengisian kuesioner dan saran serta tombol untuk melakukan login admin.

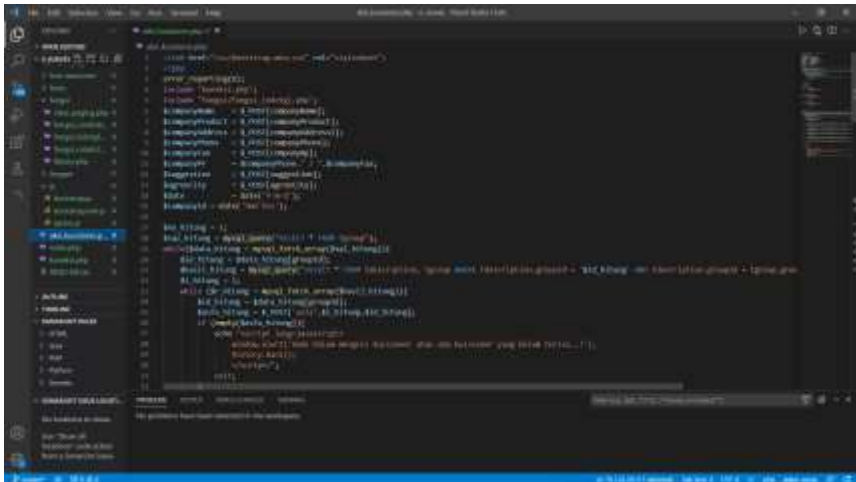
4.4.1.2 Pengujian Terhadap Koneksi Database



Gambar 28-4.22 Pengujian Koneksi Database

Pada pengujian ini, codingan koneksi *database* tidak ditemukan kesalahan yang terjadi. Codingan ini merupakan pemanggilan data dari database yang akan ditampilkan dalam sistem ini.

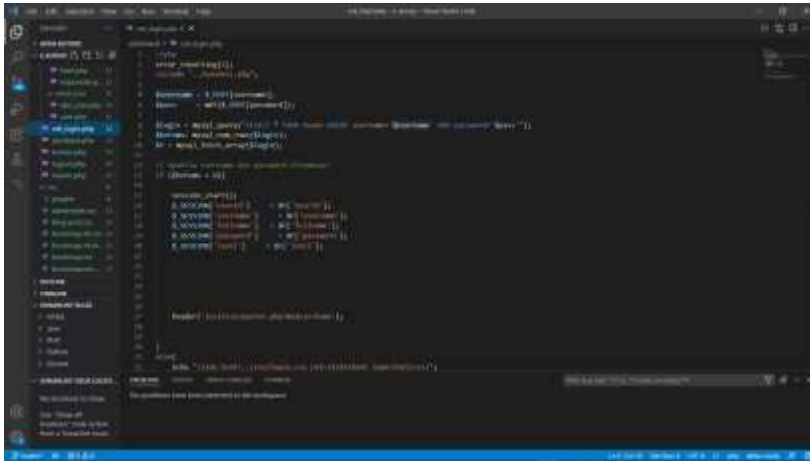
4.4.1.3 Pengujian terhadap Aksi Kuesioner



Gambar 29-4.23 Pengujian aksi Kuesioner

Pengujian ini dilakukan di Aksi Kuesioner. Codingan ini merupakan *query* untuk menginputkan data informasi pelanggan, isi kuesioner serta saran yang telah pelanggan berikan.

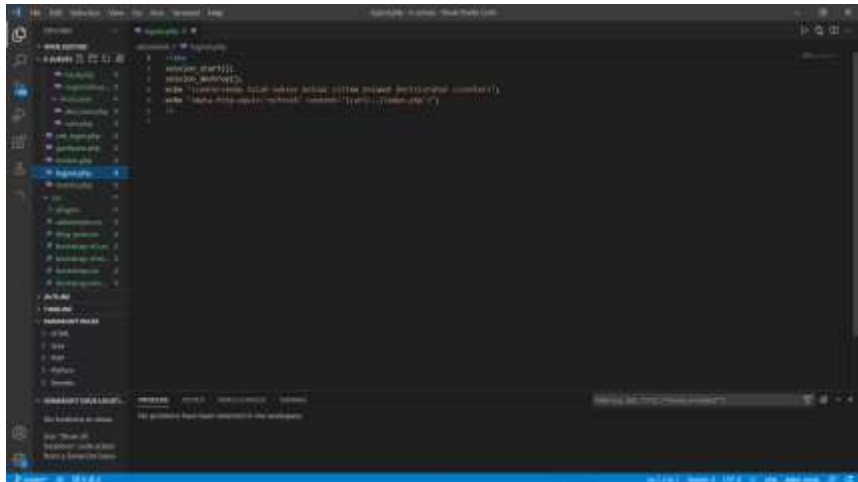
4.4.1.4 Pengujian terhadap Login



Gambar 30-4.24 Pengujian Validasi

Pada pengujian validasi juga tidak ditemukannya kesalahan. Validasi sendiri merupakan codingan untuk memvalidasi terhadap *username* serta *password* ketika melakukan *login*.

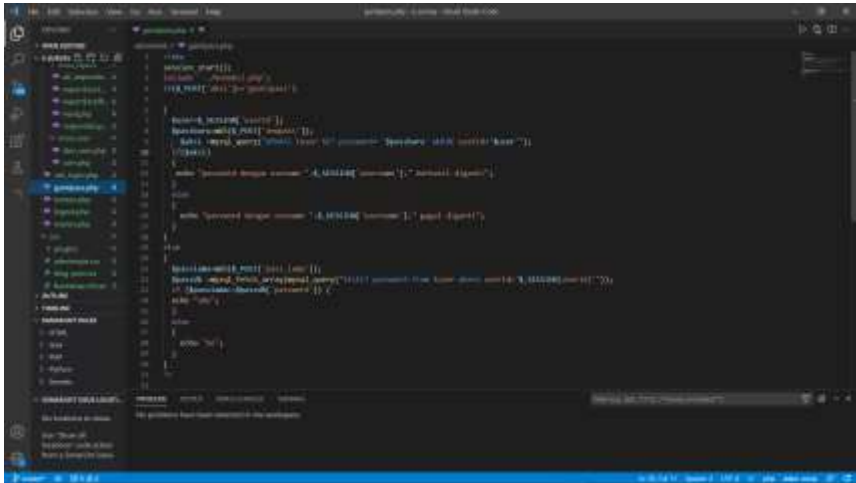
4.4.1.5 Pengujian ketika logout



Gambar 31-4.25 Pengujian Logout

Pada pengujian validasi juga tidak ditemukannya kesalahan. *Logout* sendiri merupakan codingan ketika admin selesai menggunakan sistem dan akan melakukan *logout*.

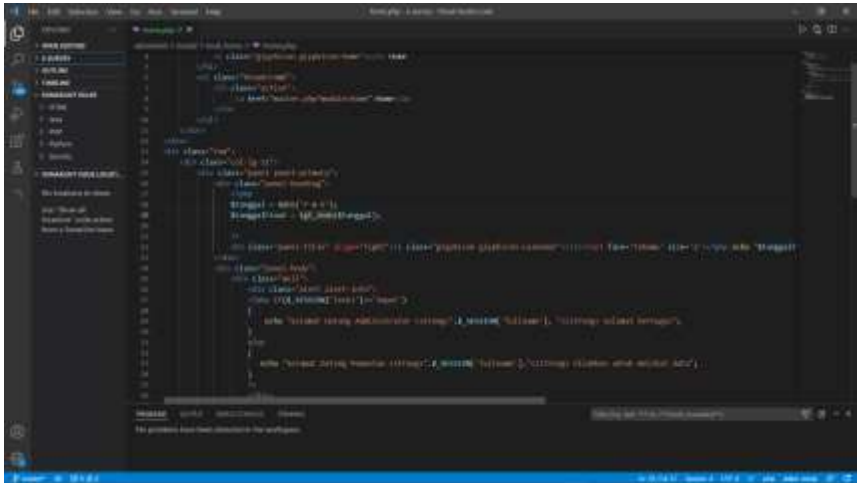
4.4.1.6 Pengujian terhadap gantipass.php



Gambar 32-4.26 Pengujian gantipass

Pada pengujian gantipass juga tidak ditemukannya kesalahan. gantipass sendiri merupakan codingan ketika admin akan melakukan pergantian *password* terhadap akunnya.

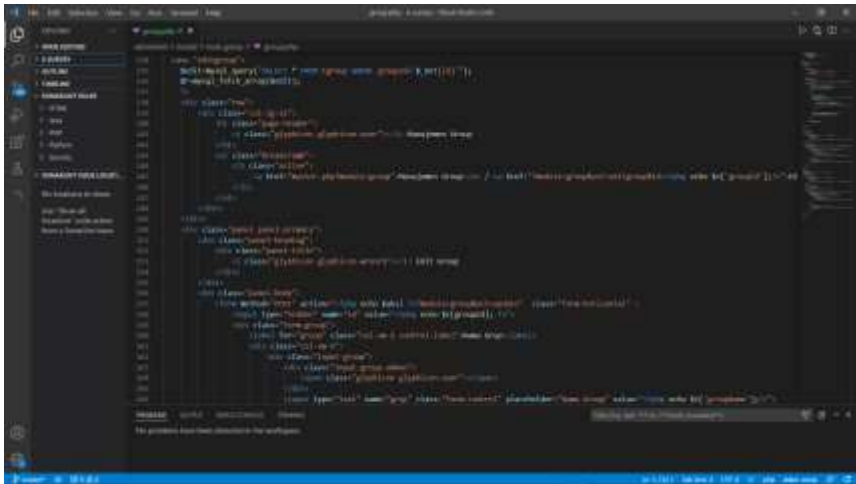
4.4.1.7 Pengujian terhadap Menu Home



Gambar 33-4.27 Pengujian Menu Home

Pada pengujian menu home juga tidak ditemukannya kesalahan. *Home* sendiri merupakan codingan ketika admin telah melakukan *login* dan akun sudah tervalidasi, maka akan muncul halaman *dashboard* admin.

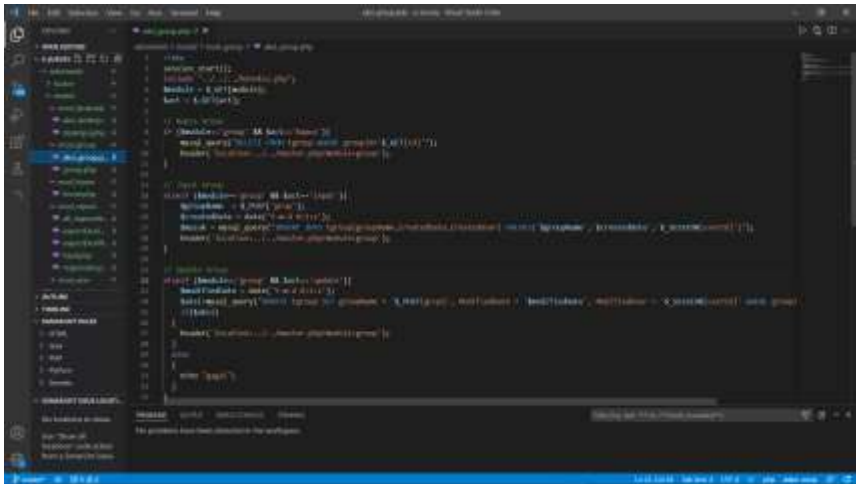
4.4.1.8 Pengujian terhadap menu Kategori



Gambar 34-4.28 Pengujian Menu Kategori

Pada pengujian menu kategori juga tidak ditemukannya kesalahan. kategori sendiri merupakan codingan ketika admin melakukan pengelolaan terhadap kategori yang akan disampaikan kepada para responden.

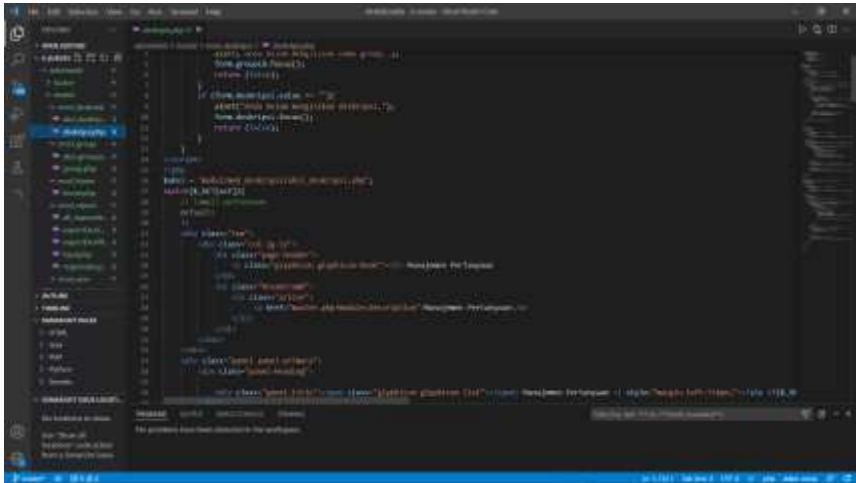
4.4.1.9 Pengujian *CRUD* Kategori



Gambar 35-4.29 Pengujian *CRUD* Kategori

Pada pengujian *CRUD* juga tidak ditemukannya kesalahan. *CRUD* kategori merupakan codingan ketika admin menambahkan, mengedit serta menghapus kategori pertanyaan dalam menu kategori .

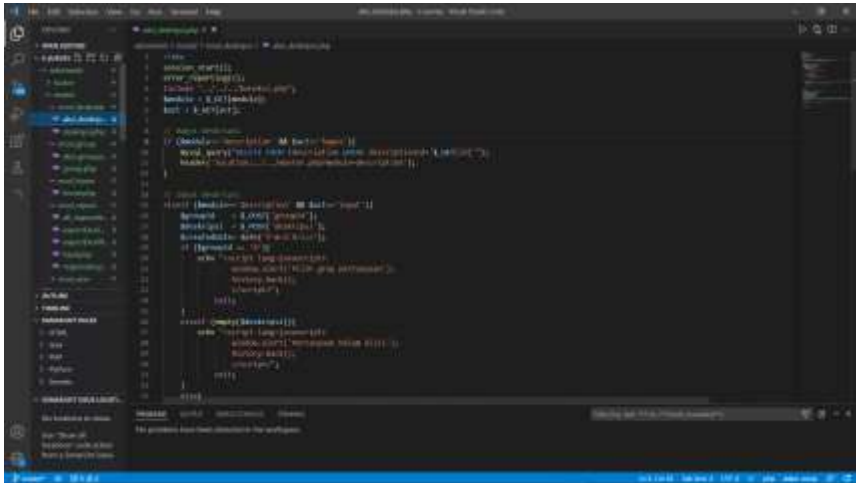
4.4.1.10 Pengujian Terhadap Menu Pertanyaan



Gambar 36-4.30 Pengujian Menu Pertanyaan

Pada pengujian menu pertanyaan juga tidak ditemukannya kesalahan. Menu Pertanyaan merupakan codingan ketika admin mengelola pertanyaan yang sedang diajukan dalam kuesioner tersebut.

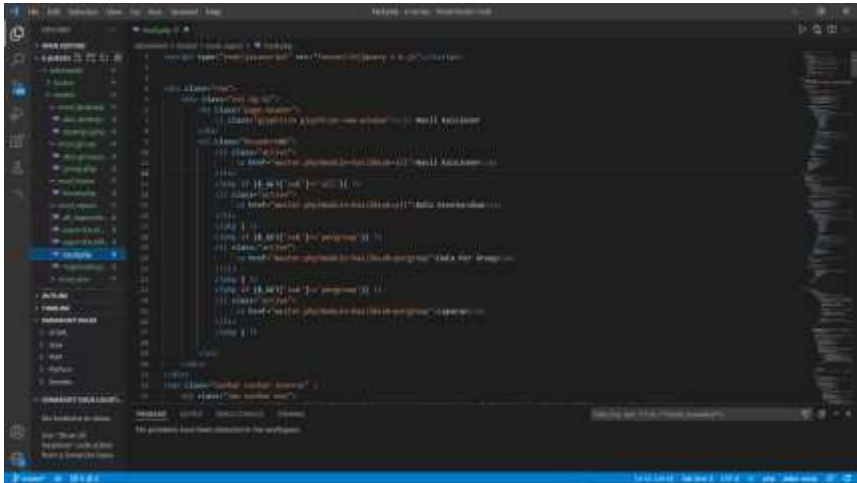
4.4.1.11 Pengujian Terhadap *CRUD* Pertanyaan



Gambar 37-4.31 Pengujian *CRUD* Pertanyaan

Pada pengujian *CRUD* pertanyaan juga tidak ditemukannya kesalahan. *CRUD* kategori merupakan codingan ketika admin menambahkan, mengedit serta menghapus pertanyaan yang terdapat dalam menu pertanyaan.

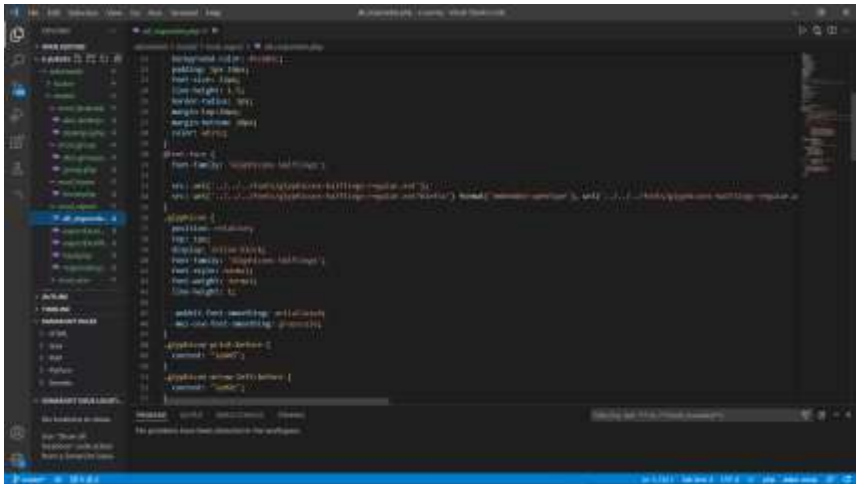
4.4.1.12 Pengujian Terhadap Menu Hasil



Gambar 38-4.32 Pengujian Menu Hasil

Pada pengujian menu hasil juga tidak ditemukannya kesalahan. Menu hasil merupakan menu admin dapat mengetahui hasil dari kuesioner tersebut. Menu hasil juga memiliki submenu yaitu: data keseluruhan, data per grup, serta laporan

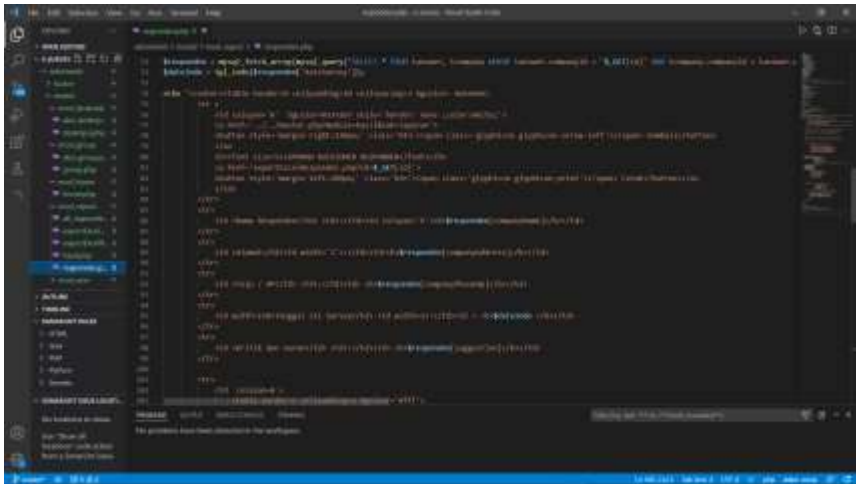
4.4.1.13 Pengujian Terhadap Hasil responden



Gambar 39-4.33 Perngujian Hasil Responden

Pada pengujian hasil responden juga tidak ditemukannya kesalahan. Hasil Reponden ini merupakan codingan untuk menampilkan hasil dari kuesioner tersebut. Hasil tersebut akan tampil di dalam submenu yaitu : Data keseluruhan serta data per kategori.

4.4.1.14 Pengujian terhadap submenu Laporan



Gambar 40-4.34 Pengujian Submenu Laporan

Pada pengujian laporan juga tidak ditemukannya kesalahan. Codingan ini bertujuan untuk menampilkan data didalam submenu laporan, yang berisi masukan serta saran dan juga nama pelanggan yang sudah mengisi kuesioner tersebut.

4.4.2 Usability Testing

Pengujian *Usability testing* dilakukan pada sistem *e-kuesioner contingency plan*. Penerapan *usability testing* teradapat di *menu* hasil pada submenu data keseluruhan hingga data per grup. Ketika pengguna sudah menginputkan hasil dari kuesioner tersebut, maka secara otomatis sistem akan menghitung berdasarkan aturan aturan yang berlaku dalam *usability testing*. Berikut submenu yang menerapkan *usability testing* :

Analisis Contingency Plan Sistem Billing Impor Menggunakan Metode User Centered Design

[illegible]

Gambar 41-35 penerapan *Usability Testing* Data Keseluruhan

The screenshot displays the Intel Resource Manager web application. The top navigation bar includes a sidebar with links for Home, Resources, Nodes, and Settings. The main content area is titled 'Resources' and features a table with columns for Node, CPU, Memory, and various storage metrics. The 'Resources' section is currently expanded, showing a detailed view of resource usage for a specific node. The 'Nodes' section is collapsed. The interface is dark-themed and includes a sidebar with navigation options like 'Home', 'Resources', 'Nodes', and 'Settings'.

Gambar 42-4.36 penerapan *Usability Testing* data per grup

KESIMPULAN

Berdasarkan pembahasan ini, dapat disimpulkan bahwa Penentuan *contingency plan* sangat dibutuhkan perusahaan mengingat manfaat dari *contingency plan* ialah untuk melakukan rencana darurat ketika sistem utama dari perusahaan mengalami gangguan. Penentuan *contingency plan* di perusahaan X menggunakan kuesioner dalam mengetahui apakah dibutuhkan *contingency plan* atau tidak di perusahaan tersebut.

Kuesioner tersebut dibuat melalui sistem kuesioner yang berbasis web sehingga memudahkan perusahaan untuk melihat hasil dari kuesioner tersebut. Manfaat kuesioner tersebut tidak hanya tentang kuesioner, namun bisa digunakan untuk mengetahui sistem yang sering mengalami gangguan sehingga pihak pengembang dapat melakukan perbaikan dengan segera. Untuk sementara, manfaat *contingency plan* belum terlaksana dikarenakan sistem di perusahaan masih berjalan dengan baik sehingga sistem *contingency plan* belum bisa digunakan.

DAFTAR PUSTAKA

- Sutanta, E. (2018). Aplikasi Penjualan Barang Perlengkap. *Jumantaka*, 1(1), 61–70. <http://jurnal.stmik-dci.ac.id/index.php/jumantaka/>
- Sulistyorini, P. (2009). Pemodelan Visual dengan Menggunakan UML dan Rational Rose. *Jurnal Teknologi Informasi DINAMIK Volume*, XIV(1), 23–29.
- Sonata, F.-. (2019). Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer. *Jurnal Komunika : Jurnal Komunikasi, Media Dan Informatika*, 8(1), 22. <https://doi.org/10.31504/komunika.v8i1.1832>
- Citra Dewi, R., Pratiwi, H., Rahmamuthi, A., Agus Petra, B., & Ramadhanu, A. (2019). Pengaruh Sistem *E-Billing* Dan Kualitas Pelayanan Terhadap Kepatuhan Wajib Pajak. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 1(2), 13–17. <https://doi.org/10.47233/jteksis.v1i2.45>

- BR Silitonga, R., Ishak, Z., & Mukhlis, M. (2019). Pengaruh ekspor, impor, dan inflasi terhadap nilai tukar rupiah di Indonesia. *Jurnal Ekonomi Pembangunan*, 15(1), 53–59. <https://doi.org/10.29259/jep.v15i1.8821>
- Ratnadi, N. M. D., & Widanaputra, A. A. G. P. (2019). Pengaruh Persepsi Kegunaan, Kemudahan Penggunaan dan Norma Subyektif pada Minat Berperilaku Penggunaan *E-Billing*. *Jurnal Ilmiah Akuntansi Dan Bisnis*, 14(2), 169. <https://doi.org/10.24843/jiab.2019.v14.i02.p03>
- Simarmata, J. (2008). Perancangan Basis Data. In Penerbit Andi, Yogyakarta.
- Rizawanti, R., Resika Arthana, I. K., & Arta Suyasa, P. W. (2019). Usability Testing Pada Aplikasi Hooki Arisan Dengan Model Pacmad Menggunakan Pendekatan Gqm. *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI)*, 8(1), 33.

