BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi sistem merupakan tahap untuk menerapkan perancangan yang telah dilakukan terhadap sistem. Tahap ini merupakan tahap dimana sistem telah siap untuk digunakan, yang terdiri dari penjelasan mengenai lingkungan implementasi, dan implementasi program. Implementasi sistem meliputi implementasi perangkat keras, implementasi perangkat lunak, implementasi basis data, dan implementasi antar muka.

4.1.1 Perangkat Keras Yang Digunakan

Perangkat keras yang digunakan dalam pembangunan sistem manajemen proyek di PT Imeco Inter Sarana memiliki spesifikasi yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras Yang Digunakan

No	Perangkat Keras	Spesifikasi
1	Prosesor	*Intel Core* i3 2,30 Ghz
2	Memori	2 GB
3	VGA Card	128 MB
4	Harddisk	320 GB
5	Monitor	Monitor 14”
6	*Mouse*	Standar
7	Printer	Standar
8	Koneksi	Kecepatan diatas 512 Mbp/s

4.1.2 Perangkat Lunak Yang Digunakan

Perangkat lunak yang digunakan dalam pembangunan sistem manajemen proyek pada PT Imeco Inter Sarana memiliki spesifikasi yang dapat dilihat pada Tabel 4.2.

Tabel 4.2 Perangkat Lunak Yang Digunakan

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 8 Pro
2	*Web Browser*	Google Chrome
3	Bahasa Pemrograman	PHP
4	DBMS	MySQL
5	*Code Editor*	Sublime Text 3
6	*UML Modeler*	Microsoft Visio 2010
7	*Web Server*	XAMPP 3.2.1

4.1.3 Implementasi Basis Data

Pembuatan basis data dilakukan dengan menggunakan aplikasi DBMS MySQL. Implementasi basis data dalam bahasa SQL adalah sebagai berikut:

1. Tabel Users

Tabel user digunakan untuk menyimpan data user yang dapat menggunakan sistem. Struktur tabel users dapat dilihat pada Tabel 4.3.

Tabel 4.3 Implementasi Tabel Users

Nama Tabel	Perintah
Users	CREATE TABLE IF NOT EXISTS `users` (`id_user` int(5) NOT NULL AUTO_INCREMENT, `username` varchar(20) NOT NULL, `password` varchar(130) NOT NULL, `nama_user` varchar(30) NOT NULL, `inisial` varchar(4) NOT NULL, `email` varchar(50) NOT NULL, `status_akun` varchar(10) NOT NULL, `status_user` varchar(15) NOT NULL, PRIMARY KEY (`id_user`), KEY `id_user` (`id_user`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

2. Tabel Klien

Tabel klien digunakan untuk menyimpan data klien yang menggunakan jasa perusahaan. Struktur tabel klien dapat dilihat pada Tabel 4.4.

Tabel 4.4 Implementasi Tabel Klien

Nama Tabel	Perintah
Klien	CREATE TABLE IF NOT EXISTS `klien` (`id_klien` int(5) NOT NULL AUTO_INCREMENT, `spk` varchar(30) NOT NULL, `nama_klien` varchar(30) NOT NULL, `telepon` varchar(12) NOT NULL, `instansi` varchar(50) NOT NULL, `tgl` date NOT NULL, `id_user` int(5) NOT NULL, PRIMARY KEY (`id_klien`), KEY `id_user` (`id_user`), CONSTRAINT `klien_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `users` (`id_user`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

3. Tabel Proyek

Tabel proyek digunakan untuk menyimpan data proyek yang dilakukan oleh manajer. Struktur tabel proyek dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi Tabel Proyek

Nama Tabel	Perintah
Proyek	CREATE TABLE IF NOT EXISTS `proyek` (`kode_proyek` varchar(15) NOT NULL, `id_klien` int(5) NOT NULL, `nama_proyek` varchar(100) NOT NULL, `biaya` float NOT NULL, `status` varchar(15) NOT NULL, `tgl` date NOT NULL, PRIMARY KEY (`kode_proyek`), KEY `id_klien` (`id_klien`), CONSTRAINT `proyek_ibfk_1` FOREIGN KEY (`id_klien`) REFERENCES `klien` (`id_klien`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4. Tabel Struktur_Kerja

Tabel struktur_kerja digunakan untuk menyimpan data kegiatan struktur kerja pada suatu proyek. Struktur tabel struktur_kerja dapat dilihat pada Tabel 4.6.

Tabel 4.6 Implementasi Tabel Struktur_Kerja

Nama Tabel	Perintah
Struktur_Kerja	CREATE TABLE IF NOT EXISTS `struktur_kerja` (`id_kerja` int(5) NOT NULL AUTO_INCREMENT, `kode_proyek` varchar(15) NOT NULL, `id_pekerjaan` int(5) NOT NULL, `nama_kegiatan` varchar(100) NOT NULL, `mulai` date NOT NULL, `selesai` date NOT NULL, `supervisor` int(5) NOT NULL, `biaya` float NOT NULL, PRIMARY KEY (`id_kerja`), KEY `id_pekerjaan` (`id_pekerjaan`), KEY `kode_proyek` (`kode_proyek`), CONSTRAINT `struktur_kerja_ibfk_2` FOREIGN KEY (`kode_proyek`) REFERENCES `proyek` (`kode_proyek`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `struktur_kerja_ibfk_1` FOREIGN KEY (`id_pekerjaan`) REFERENCES `pekerjaan` (`id_pekerjaan`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=46 ;

5. Tabel Detail_Kerja

Tabel detail_kerja digunakan untuk menyimpan data penjadwalan berdasarkan struktur kerja pada suatu proyek. Struktur tabel detail_kerja dapat dilihat pada Tabel 4.7.

Tabel 4.7 Implementasi Tabel Detail_Kerja

Nama Tabel	Perintah
Detail_Kerja	CREATE TABLE IF NOT EXISTS `detail_kerja` (`id_detail` int(5) NOT NULL AUTO_INCREMENT, `id_kerja` int(5) NOT NULL, `id_material` int(5) NOT NULL, `jumlah` int(11) NOT NULL, PRIMARY KEY (`id_detail`), KEY `id_kerja` (`id_kerja`), KEY `id_material` (`id_material`), CONSTRAINT `detail_kerja_ibfk_2` FOREIGN KEY (`id_material`) REFERENCES `material` (`id_material`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `detail_kerja_ibfk_1` FOREIGN KEY (`id_kerja`) REFERENCES `struktur_kerja` (`id_kerja`) ON DELETE CASCADE ON UPDATE CASCADE,)
Detail_Kerja	ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

6. Tabel Laporan

Tabel laporan digunakan untuk menyimpan data laporan berdasarkan jadwal suatu proyek yang dilakukan oleh supervisor. Struktur tabel laporan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Implementasi Tabel Laporan

Nama Tabel	Perintah
Laporan	CREATE TABLE IF NOT EXISTS `laporan` (`id_laporan` int(5) NOT NULL AUTO_INCREMENT, `id_kerja` int(5) NOT NULL, `catatan` text NOT NULL, `status` varchar(15) NOT NULL, PRIMARY KEY (`id_laporan`), KEY `id_kerja` (`id_kerja`), CONSTRAINT `laporan_ibfk_1` FOREIGN KEY (`id_kerja`) REFERENCES `struktur_kerja` (`id_kerja`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=93 ;

7. Tabel Identifikasi

Tabel identifikasi digunakan untuk menyimpan data identifikasi resiko berdasarkan proyek yang dilaksanakan. Struktur tabel identifikasi dapat dilihat pada Tabel 4.9.

Tabel 4.9 Implementasi Tabel Identifikasi

Nama Tabel	Perintah
Identifikasi	CREATE TABLE IF NOT EXISTS `identifikasi` (`id_identifikasi` int(5) NOT NULL AUTO_INCREMENT, `kode_resiko` varchar(5) NOT NULL, `jenis_resiko` varchar(30) NOT NULL, `identifikasi` text NOT NULL, PRIMARY KEY (`id_identifikasi`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;

8. Tabel Penanganan

Tabel penanganan digunakan untuk menyimpan data penanganan resiko berdasarkan identifikasi proyek yang mungkin terjadi. Struktur tabel penanganan dapat dilihat pada Tabel 4.10.

Tabel 4.10 Implementasi Tabel Penanganan

Nama Tabel	Perintah
Penanganan	CREATE TABLE IF NOT EXISTS `penanganan` (`id_penanganan` int(5) NOT NULL AUTO_INCREMENT, `id_tingkep` int(5) NOT NULL, `tingkat_resiko` varchar(15) NOT NULL, `penanganan` text NOT NULL, PRIMARY KEY (`id_penanganan`), KEY `id_tingkep` (`id_tingkep`), CONSTRAINT `penanganan_ibfk_1` FOREIGN KEY (`id_tingkep`) REFERENCES `tingkat_kepentingan` (`id_tingkep`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;

9. Tabel Tingkat_Kepentingan

Tabel tingkat kepentingan digunakan untuk menyimpan data tingkat kepentingan resiko berdasarkan identifikasi proyek yang mungkin terjadi. Struktur tabel tingkat kepentingan dapat dilihat pada Tabel 4.11.

Tabel 4.11 Implementasi Tabel Tingkat_Kepentingan

Nama Tabel	Perintah
Tingkat_Kepentingan Tingkat_Kepentingan	CREATE TABLE IF NOT EXISTS `tingkat_kepentingan` (`id_tingkep` int(5) NOT NULL AUTO_INCREMENT, `kode_proyek` varchar(15) NOT NULL, `id_identifikasi` int(5) NOT NULL, `probabilitas` tinyint(2) NOT NULL, `dampak` tinyint(2) NOT NULL, `tingkep` tinyint(2) NOT NULL, PRIMARY KEY (`id_tingkep`), KEY `kode_proyek` (`kode_proyek`), KEY `id_identifikasi` (`id_identifikasi`), CONSTRAINT `tingkat_kepentingan_ibfk_2` FOREIGN KEY (`id_identifikasi`) REFERENCES `identifikasi` (`id_identifikasi`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `tingkat_kepentingan_ibfk_1` FOREIGN KEY (`kode_proyek`) REFERENCES `proyek` (`kode_proyek`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;

10. Tabel CPM

Tabel CPM digunakan untuk memperlihatkan jalur kritis proyek berdasarkan jadwal pelaksanaan proyek. Struktur tabel CPM dapat dilihat pada Tabel 4.12.

Tabel 4.12 Implementasi Tabel CPM

Nama Tabel	Perintah
CPM	CREATE TABLE IF NOT EXISTS `cpm` (`id_cpm` int(5) NOT NULL AUTO_INCREMENT, `kode_proyek` varchar(15) NOT NULL, `kode_kegiatan` varchar(1) NOT NULL, `kode_pendahulu` varchar(8) NOT NULL, `waktu` int(3) NOT NULL, PRIMARY KEY (`id_cpm`), KEY `kode_proyek` (`kode_proyek`), CONSTRAINT `cpm_ibfk_1` FOREIGN KEY (`kode_proyek`) REFERENCES `proyek` (`kode_proyek`) ON DELETE CASCADE ON UPDATE CASCADE,) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=34 ;

11. Tabel Material

Tabel material digunakan untuk menyimpan data material berdasarkan kegiatan dari struktur kerja. Struktur tabel material dapat dilihat pada Tabel 4.13.

Tabel 4.13 Implementasi Tabel Material

Nama Tabel	Perintah
Material	CREATE TABLE IF NOT EXISTS `material` (`id_material` int(5) NOT NULL AUTO_INCREMENT, `material` varchar(50) NOT NULL, `satuan` varchar(10) NOT NULL, `harga` float NOT NULL, PRIMARY KEY (`id_material`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=19 ;

12. Tabel Pekerjaan

Tabel pekerjaan digunakan untuk menyimpan data pekerjaan berdasarkan struktur kerja. Struktur tabel pekerjaan dapat dilihat pada Tabel 4.14.

Tabel 4.14 Implementasi Tabel Pekerjaan

Nama Tabel	Perintah
Pekerjaan	CREATE TABLE IF NOT EXISTS `pekerjaan` (`id_pekerjaan` int(5) NOT NULL AUTO_INCREMENT, `pekerjaan` varchar(50) NOT NULL, `bobot_kegiatan` float NOT NULL, PRIMARY KEY (`id_pekerjaan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=14 ;

4.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangunan dan pengkodeannya dalam bentuk *file* program. Adapun untuk tampilan implementasi sistem manajemen proyek di PT Imeco Inter Sarana akan dilampirkan pada halaman lampiran. Berikut ini adalah implementasi antarmuka sistem manajemen proyek di PT Imeco Inter Sarana, yang terbagi atas antarmuka admin, antarmuka manajer, dan antarmuka supervisor.

Implementasi antarmuka untuk admin dapat dilihat pada Tabel 4.15 berikut.

Tabel 4.15 Implementasi Antarmuka Admin

No	Menu	Deskripsi	Nama File
1	Halaman *Login*	Digunakan sebagai halaman *login* bagi pengguna sistem	Login.php
2	Halaman Utama	Digunakan sebagai halaman utama pada Admin	Home.php
3	Halaman User	Digunakan sebagai halaman pengolahan users	V_user.php
4	Halaman Master Material	Digunakan sebagai halaman pengolahan master material yang digunakan dalam proyek	V_material.php
5	Halaman Master Identifikasi	Digunakan sebagai halaman pengolahan master identifikasi proyek yang mungkin terjadi	V_identifikasi.php

Implementasi antarmuka untuk manajer dapat dilihat pada Tabel 4.16 berikut.

Tabel 4.16 Implementasi Antarmuka Manajer

No	Menu	Deskripsi	Nama File
1	Halaman *Login*	Digunakan sebagai halaman *login* bagi pengguna sistem	Login.php
2	Halaman Utama	Digunakan sebagai halaman utama pada Manajer	Home.php
3	Halaman Master Pekerjaan	Digunakan sebagai halaman pengolahan kegiatan pekerjaan yang dilakukan pada suatu proyek	V_pekerjaan.php
4	Halaman Klien	Digunakan sebagai halaman pengolahan klien	V_client.php
5	Halaman Proyek	Digunakan sebagai halaman pengolahan proyek	V_proyek.php
6	Halaman Manajemen Proyek	Digunakan sebagai halaman manajemen proyek yang sebelumnya memilih proyek yang akan dibuka	Cari.php
7	Halaman Laporan Pekerjaan	Digunakan sebagai halaman melihat laporan pekerjaan yang dimasukkan oleh supervisor	V_jadwalkerja.php
8	Halaman Jadwal	Digunakan sebagai halaman melihat jadwal, kegiatan, dan hasil analisa menggunakan metode CPM	V_cpm.php
9	Halaman Biaya	Digunakan sebagai halaman melihat estimasi jadwal dan biaya suatu proyek berdasarkan analisa menggunakan metode EVM	V_evm.php
10	Halaman Tingkat Kepentingan	Digunakan sebagai halaman pengolahan resiko berdasarkan identifikasi dan level resiko yang mungkin akan terjadi pada suatu proyek	V_tingkep.php
11	Halaman Penanganan	Digunakan sebagai halaman penanganan berdasarkan identifikasi dan level resiko suatu proyek	Penanganan.php
12	Halaman Pencapaian Proyek	Digunakan sebagai halaman dalam *progress* pekerjaan	Pencapaian.php

Implementasi antarmuka untuk supervisor dapat dilihat pada Tabel 4.17 berikut.

Tabel 4.17 Implementasi Antarmuka Supervisor

No	Menu	Deskripsi	Nama File
1	Halaman Login	Digunakan sebagai halaman *login* bagi pengguna sistem	Login.php
2	Halaman Utama	Digunakan sebagai halaman utama pada Manajer	Home.php
3	Halaman Manajemen Proyek	Digunakan sebagai halaman manajemen proyek yang sebelumnya memilih proyek yang akan dibuka	Cari.php
4	Halaman Struktur Kerja	Digunakan sebagai halaman untuk pengolahan struktur kerja proyek	V_strukturkerja.php
5	Halaman Laporan Pekerjaan	Digunakan sebagai halaman melihat laporan pekerjaan yang dimasukkan oleh supervisor	V_jadwalkerjax.php
6	Halaman Pencapaian Proyek	Digunakan sebagai halaman dalam *progress* pekerjaan	Pencapaian.php

4.2 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan dan kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Pengujian yang dilakukan yaitu menggunakan pengujian *white box,* dan pengujian *black box.*

4.2.1 Skenario Pengujian *White Box*

Pengujian *white box* digunakan untuk menguji sistem. Pengujian white box untuk meyakinkan semua perintah dan kondisi dieksekusi secara minimal. Skenario pengujian *white box* dapat dilihat pada Tabel 4.18.

Tabel 4.18 Skenario Pengujian *White Box*

Fungsional Uji	Butir Uji	Jenis Pengujian
Algoritma	Uji algoritma menggunakan *Critical Path Method*	*White Box*
	Uji algoritma menggunakan metode *Earned Value Management*	
	Uji algoritma menggunakan metode *Probability Impact and Matrix*	

4.2.1.1 Kasus dan Hasil Pengujian *White Box*

Pengujian *white box* menggunakan dua *tools* atau alat yaitu *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *graph matrix* yang digunakan untuk menggenerasi *flow graph*. Berikut merupakan hasil pengujian *white box*.

1. Pengujian Algoritma Metode *Critical Path Method*

Berikut merupakan langkah-langkah tahap pengujian algoritma metode *Critical Path Method* (CPM).

a. Mengubah *pseudocode* menjadi *flowchart* pengujian. Berikut merupakan *pseudocode* algoritma metode *Critical Path Method* yang diuji pada Tabel 4.19.

Tabel 4.19 *Pseudocode* Algoritma Metode *Critical Path Method*

No	*Pseudocode*
1	`$no = 0;`
	`$s = 'A';`
	`$no_j=1;`
2	`foreach ($cpm as $cp) {`
	`$i++;`
	`$kp [$cp->kode_kegiatan]= $cp->kode_pendahulu;`
	`$es[$cp->kode_kegiatan]= $cp->waktu;`
	`$no_node[$cp->no_node]= $cp->waktu;`
	`$id_cp [$cp->kode_kegiatan]= $cp->id_cpm;`
	`$kopen[$cp->kode_pendahulu]= $cp->kode_pendahulu;`
	`$i_arr[$cp->kode_kegiatan]= $cp->i_node;`
	`$j_arr[$cp->kode_kegiatan]= $cp->j_node;`
	`}`
3	`$kon_ef=$cpm_limit->kode_kegiatan;`
	`foreach ($cpm as $cp) {`
4	`if($cp->kode_pendahulu=="-")`
	`{`
	`$n_es=0;`
	`$n_ef=0+$cp->waktu;`
	`$atas=0;`
	`$bawah=0;`
	`$kont='-';`
5	`}`
6	`else`
	`{`
7	`$kode_sebelumnya=$kont;`
	`$tampung=$kp[$cp->kode_kegiatan];`
	`$n_esx=$tampung;`
	`list($x1,$x2)= explode(",", $tampung);`
8	`if(strlen($tampung)>1)`
	`{`
9	`if($es[$x1] > $es[$x2])`
	`{`

No	*Pseudocode*
	$n_es=$es[$x1]; $n_ef=$es[$x1]+$cp->waktu; $kont=$x1;
10	}

11	else
	{
	$n_es=$es[$x2];
	$n_ef=$es[$x2]+$cp->waktu;
	$kont=$x2;
	}
	$atas=$es[$x1];
	$bawah=$es[$x2];

12	}

13	else
	{
	$n_es=$es[$tampung];
	$atas=$es[$tampung];
	$bawah=$es[$tampung];
	$n_ef=$es[$tampung]+$cp->waktu;
	$kont=$tampung;
	}

14	}

15	if($kode_sebelumnya==$kont)
	{
	$jum=$jum;
	$jum_atas=$jum_atas;
	$jum_bawah=$jum_bawah;
	}

16	else
	{
	$jum=$n_es+$jum;
	$jum_atas=$jum_atas+$atas;
	$jum_bawah=$jum_bawah+$bawah;
	}

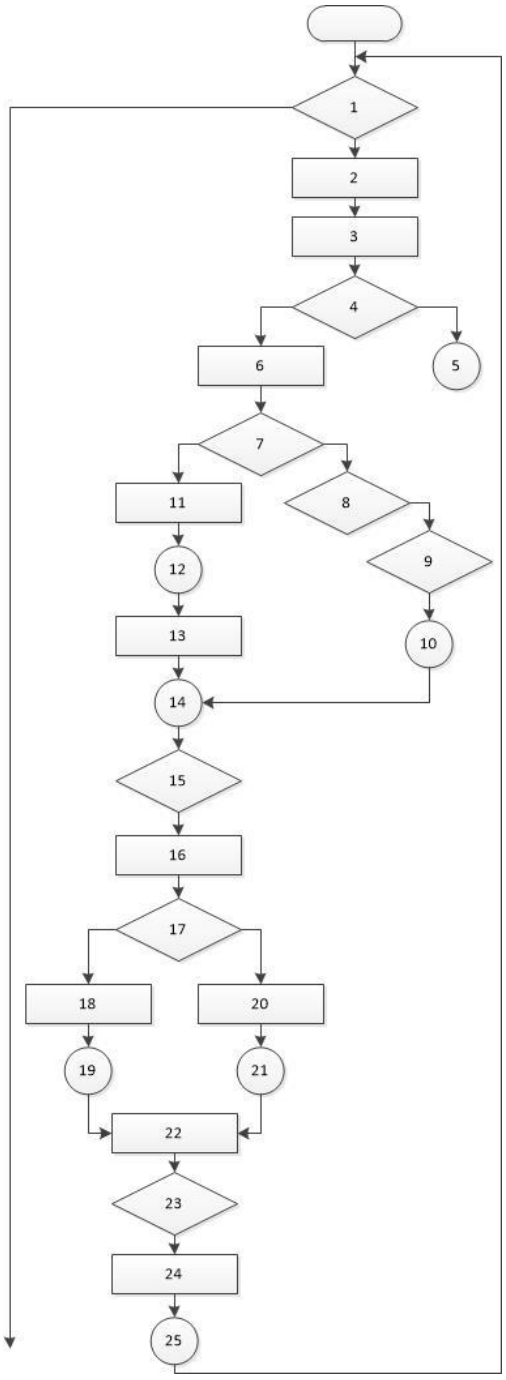
17	if($jum==0)
	{
	$jum_ef=$es[$kon_ef];
	}

18	else
	{
	$jum_ef=$jum;

19	}
20	$jalus_kritis[]= $kont; $hasil_es[$cp->kode_kegiatan]= $jum; $hasil_ef[$kon_ef]= $jum_ef; $hasil_atas[$cp->kode_kegiatan]= $jum_atas; $hasil_bawah[$cp->kode_kegiatan]= $jum_bawah; $kon_ef=$cp->kode_kegiatan; $total_hari=$jum_ef+$total_hari; }; $ls=$jum+1;
21	foreach ($cpm_desc as $cp_d) {
22	$i++; $ls_sebelumnya=$ls; if($i==1)

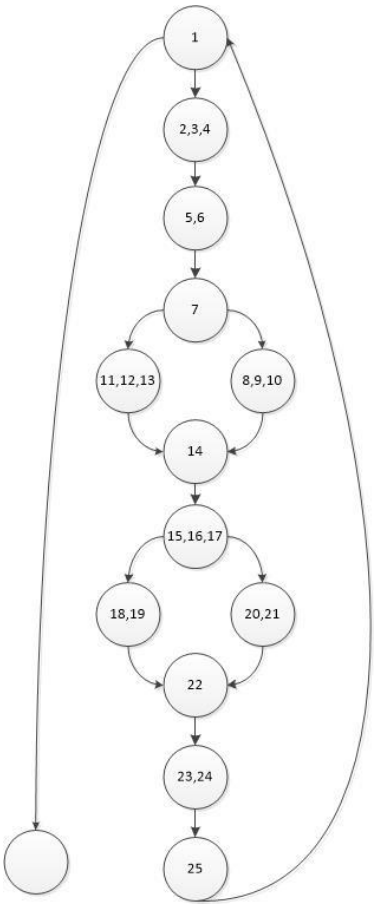
No	*Pseudocode*
	{
23	$ls=($jum+1)-$cp_d->waktu;
	}
24	else
	{
	$ls=$ls_sebelumnya-$cp_d->waktu;
25	}
	$ls=$ls_sebelumnya-$cp_d->waktu;
	$hasil_ls[$cp_d->kode_kegiatan]= $ls;
26	};

Setelah mengetahui algoritma, selanjutnya mengubah menjadi *flowchart* pengujian yang sesuai dengan *pseudocode* pada Tabel 4.19. Adapun berikut *flowchart* dari algoritma metode CPM dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Flowchart* Algoritma Metode *Critical Path Method*

b. Selanjutnya menentukan *flowgraph* dari pengujian *flowchart* pengujian yang sesuai dengan *pseudocode* pada Tabel 4.19 sebelumnya. Adapun berikut *flowgraph* algoritma metode *Critical Path Method* dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Flowgraph* Algoritma Metode *Critical Path Method*

c. Tahap pengujian dalam menentukan *region*, *cyclomatic complexity*, *independent path*, *graph matrix*, dan *predicate node*. Berikut merupakan penjabarannya.

1) Region adalah area yang dibatasi oleh *edge* dan node disebut region, area diluar *graph* juga dihitung sebagai region. Rumus jumlah region adalah jumlah kurva tertutup ditambah 1 atau region terluar pada *flowgraph* pada Gambar 4.2.

   Jumlah region = 3 + 1 = 4

   Terdapat empat region dalam *flowgraph* pada Gambar 4.2.

2) Kompleksitas siklomatis adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Rumus kompleksitas siklomatis (*cyclomatic complexity*) adalah

V(G) = E – N + 2. Dimana E adalah jumlah busur pada *flowgraph*, sedangkan N merupakan jumlah simpul pada *flowgraph*.

V(G) = 16 – 14 + 2 = 4

Terdapat empat kompleksitas siklomatis dalam *flowgraph* pada Gambar 4.2.

3) Jalur *independent* adalah jalur yang terdapat dalam program yang memperkenalkan setidaknya satu rangkaian pernyataan proses atau kondisi baru. Sebuah *independent path* harus bergerak setidaknya sepanjang satu sisi yang belum ditelusuri sebelum jalur tersebut didefinisikan. Berdasarkan *flowgraph* pada Gambar 4.2 didapatkan jalur *independent* yang ada sebagai berikut:

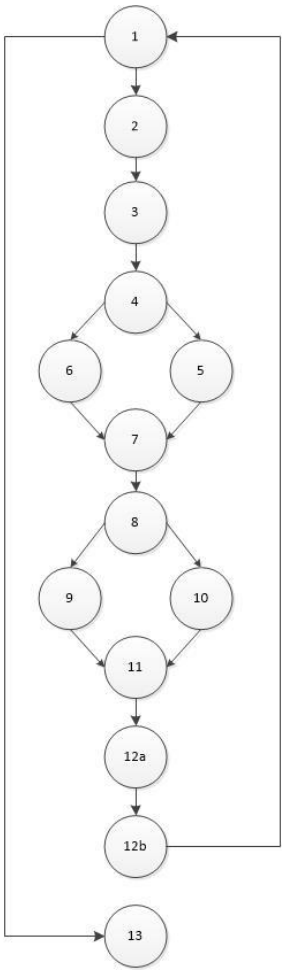
Path 1 : 1-26

Path 2 : 1-2-3-4-5-6-7-11-12-13-14-15-16-17-18-19-22-23-24-25-1-26

Path 3 : 1-2-3-4-5-6-7-8-9-10-14-15-16-17-20-21-22-23-24-25-1-26

Path 4 : 1-2-3-4-5-6-7-8-9-10-14-15-16-17-20-21-22-23-24-25-1-2-3-4-5-6-7-11-12-13-14-15-16-17-18-19-22-23-24-25-1-26

Selanjutnya penomoran ulang untuk *flowgraph* yang sesuai pada *flowgraph pada* Gambar 4.2 sebelumnya. Adapun berikut merupakan *flowgraph* algoritma *Critical Path Method* penomoran ulang yang dapat dilihat pada Gambar 4.3.



Gambar 4.3 *Flowgraph* Penomoran Ulang Algoritma *Critical Path Method*

4) *Graph matrix* adalah matrik berbentuk segiempat sama sisi, dimana jumlah baris dan kolom sama dengan jumlah node, dan identifikasi baris dan kolom sama dengan identifikasi node, serta isi data adalah keberadaan penghubung antar node (*edges*). Berdasarkan penomoran ulang pada *flowgraph* pada Gambar 4.3, adapun berikut *graph matrix* pada algoritma *critical path method* dapat dilihat pada Tabel 4.20.

Tabel 4.20 *Graph Matrix* Algoritma Metode *Critical Path Method*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12a**	**12b**	**13**	**n(E)-1**
1		1												1	1
2			1												0
3				1											0
4					1	1									1
5							1								0
6							1								0
7								1							0
8									1	1					1

	1	2	3	4	5	6	7	8	9	10	11	12a	12b	13	n(E)-1
9											1				0
10											1				0
11												1			0
12a													1		0
12b	1														0
13															0
														Sum(E)+1	3+1=4

5) *Predicate node* adalah proses yang membentuk dua cabang atau lebih atau node yang mempunyai kondisi berdasarkan *flowgraph* Gambar 4.2. *Predicate node* memiliki rumus yaitu V(G) = Jumlah *predicate node* + 1.
V(G) = 3 + 1 = 4
Predicate node pada algoritma *critical path method* adalah 4.

2. Pengujian Algoritma Metode *Earned Value Management*

Berikut merupakan langkah-langkah tahap pengujian algoritma metode *Earned Value Management* (EVM).

a. Mengubah *pseudocode* menjadi *flowchart* pengujian. Berikut merupakan *pseudocode* algoritma metode *Earned Value Management* yang diuji pada Tabel 4.21.

Tabel 4.21 *Pseudocode* Algoritma Metode *Earned Value Management*

No	*Pseudocode*

1	`$hari_kerja=6;//senin sampe sabtu`
	`$minggu=ceil($durasi/$hari_kerja);`
	`$n=0;`

2	`foreach ($belum as $b) {`
	`$n++;`
	`$belum_sel=$b->mulai;`
	`}`

3	`$week = ((abs(strtotime ($tgl_mulai_proyek) - strtotime ($belum_sel)))/(60*60*24));`
	`$minggu_ke=ceil($week/$hari_kerja);`
	`$belum_selesai=$n;`

4	`if($belum)`
	`{`
	`$minggu_ke=$minggu_ke;`
	`}`

5	`else`
	`{`
	`$week = ((abs(strtotime ($tgl_mulai_proyek) - strtotime ($tgl_selesai)))/(60*60*24));`
	`$minggu_ke=ceil($week/$hari_kerja)+1;`
	`}`

6	`for($i=1;$i<=$minggu;$i++)`
	`{`
	`$bac=$pro->biaya;`

No	*Pseudocode*
	$persentase=100/$minggu; $persentase_telat_perhari=$persentase/7; $persentase_belum_selesai=$belum_selesai*$persentase_telat_perhari; $persentase rencana=$persentase*$i; $persentase_perminggu=$persentase*$i;

7	if($i<$minggu_ke)
	{
	$persentase_perminggu=$persentase*$i;
	}

8	else
	{
	$persentase_perminggu=(($persentase*$i)-
	$persentase_belum_selesai);
	}

	$tot_minggu[]=$i; $rows_pv[]=(float)$persentase_rencana; }

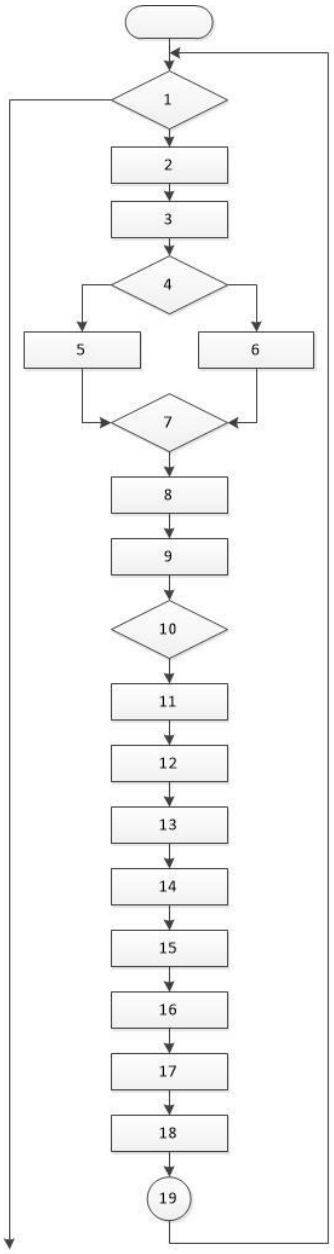
9	for($i=1;$i<=$minggu_ke;$i++)
	{
	$bac=$pro->biaya;
	$persentase=100/$minggu;
	$persentase_telat_perhari=$persentase/7;
	$persentase_belum_selesai=$belum_selesai*$persentase_telat_perhari;
	$persentase_rencana=$persentase*$i;
	$persentase_perminggu=$persentase*$i;
	$PV=($persentase_perminggu/100)*$bac;
	$AC=($persentase_perminggu/100)*$bac;

10	if($i<$minggu_ke)
	{
	$persentase_perminggu=$persentase*$i;
	}

11	else
	{
	$persentase_perminggu=(($persentase*$i)-
	$persentase_belum_selesai);
	}

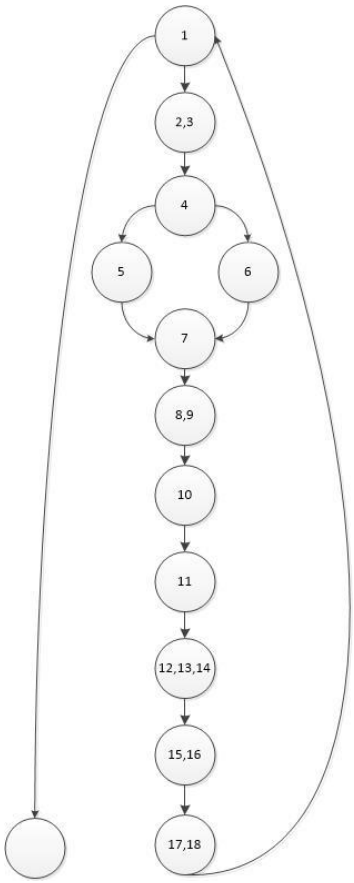
12	$EV=($persentase_perminggu/100)*$bac;
13	$SV=$EV-$PV;
14	$CV=$EV-$AC;
15	$SPI=$EV/$PV;
16	$CPI=$EV/$AC;
17	$ETC=ceil($OD/$SPI);
18	$EAC=$bac/$CPI;
19	}

Setelah mengetahui algoritma, selanjutnya mengubah menjadi *flowchart* pengujian yang sesuai dengan *pseudocode* pada Tabel 4.21. Adapun berikut merupakan *flowchart* dari algoritma metode EVM dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Flowchart* Algoritma Metode *Earned Value Management*

b. Selanjutnya menentukan *flowgraph* dari pengujian *flowchart* pengujian yang sudah dilakukan sebelumnya yang sesuai dengan *pseudocode* pada Tabel 4.21. Adapun berikut merupakan *flowgraph* algoritma metode *Critical Path Method* dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Flowgraph* Algoritma Metode *Earned Value Management*

c. Tahap pengujian dalam menentukan *region*, *cyclomatic complexity*, *independent path*, *graph matrix*, dan *predicate node*. Berikut merupakan penjabarannya.

1) Region adalah area yang dibatasi oleh *edge* dan node disebut region, area diluar *graph* juga dihitung sebagai region. Rumus jumlah region adalah jumlah kurva tertutup ditambah 1 atau region terluar pada *flowgraph*. Jumlah region sesuai dengan *pseudocode* pada Tabel 4.21.

   Jumlah region = 2 + 1 = 3

   Terdapat tiga region dalam *flowgraph* pada Gambar 4.5.

2) Kompleksitas siklomatis adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Rumus kompleksitas siklomatis (*cyclomatic complexity*) adalah V(G) = E – N + 2. Dimana E adalah jumlah busur pada *flowgraph*, sedangkan N merupakan jumlah simpul pada *flowgraph*. Berikut

merupakan hasil kompleksitas siklomatis berdasarkan *flowgraph* pada Gambar 4.5.

V(G) = 14 – 13 + 2 = 3

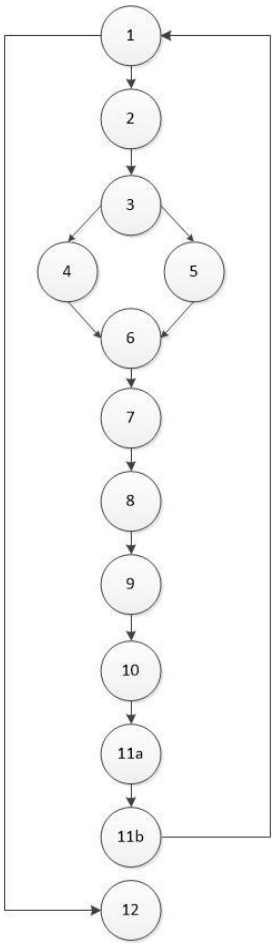
3) Jalur *independent* adalah jalur yang terdapat dalam program yang memperkenalkan setidaknya satu rangkaian pernyataan proses atau kondisi baru. Sebuah *independent path* harus bergerak setidaknya sepanjang satu sisi yang belum ditelusuri sebelum jalur tersebut didefinisikan. Berdasarkan *flowgraph* pada Gambar 4.5 didapatkan jalur *independent* yang ada sebagai berikut:

   Path 1 : 1-19

   Path 2 : 1-2-3-4-5-7-8-9-10-11-12-13-14-15-16-17-18-1-19

   Path 3 : 1-2-3-4-6-7-8-9-10-11-12-13-14-15-16-17-18-1-19

   Selanjutnya penomoran ulang untuk *flowgraph* yang sesuai pada *flowgraph pada* Gambar 4.5 sebelumnya. Berikut merupakan *flowgraph* algoritma *Earned Value Management* pada gambar 4.6.



Gambar 4.6 *Flowgraph* Penomoran Ulang Algoritma EVM

4) *Graph matrix* adalah matrik berbentuk segiempat sama sisi, dimana jumlah baris dan kolom sama dengan jumlah node, dan identifikasi baris dan kolom sama dengan identifikasi node, serta isi data adalah keberadaan penghubung antar node (*edges*). Berdasarkan penomoran ulang pada *flowgraph* pada Gambar 4.6, adapun berikut *graph matrix* pada algoritma *earned value management* dapat dilihat pada Tabel 4.22.

Tabel 4.22 *Graph Matrix* Algoritma Metode *Earned Value Management*

	1	2	3	4	5	6	7	8	9	10	11a	11b	12	n(E)-1
1		1											1	1
2			1											0
3				1	1									1
4						1								0
5						1								0
6							1							0
7								1						0
8									1					0

	1	2	3	4	5	6	7	8	9	10	11a	11b	12	n(E)-1
9										1				0
10											1			0
11a												1		0
11b	1													0
12														0
												Sum(E)+1		2+1=3

5) *Predicate node* adalah proses yang membentuk dua cabang atau lebih atau node yang mempunyai kondisi berdasarkan *flowgraph* Gambar 4.5. *Predicate node* memiliki rumus yaitu V(G) = Jumlah *predicate node* + 1.

V(G) = 2 + 1 = 3

Predicate node pada algoritma *earned value management* adalah 3.

3. Pengujian Algoritma Metode *Probability Impact and Matrix*

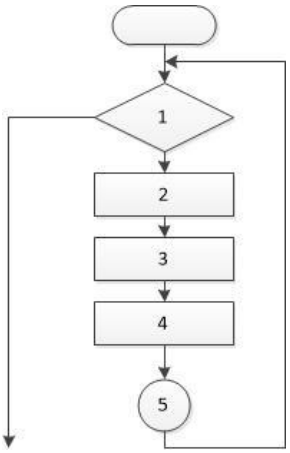
Berikut merupakan langkah-langkah tahap pengujian algoritma metode *Probability Impact and Matrix* (PIM).

a. Mengubah *pseudocode* menjadi *flowchart* pengujian. Berikut merupakan *pseudocode* algoritma metode *Probability Impact and Matrix* yang diuji pada Tabel 4.23.

Tabel 4.23 *Pseudocode* Algoritma Metode *Probability Impact and Matrix*

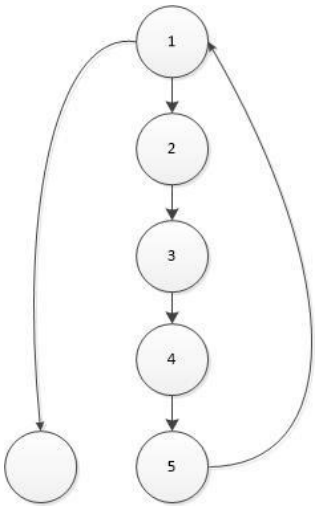
No	*Pseudocode*
1	`$("#dampak").change(function() {`
2	`var dam = $(#dampak).val();`
3	`var pro = $("#kemungkinan").val();`
4	`var risk=dam*pro;`
5	`$("#tingkep").val(risk);`
6	`});`

Setelah mengetahui algoritma, selanjutnya mengubah menjadi *flowchart* pengujian yang sesuai dengan *pseudocode* pada Tabel 4.23. Adapun berikut merupakan *flowchart* dari algoritma metode PIM yang dapat dilihat pada Gambar 4.7.



Gambar 4.7 *Flowchart* Algoritma Metode *Probability Impact and Matrix*

b. Selanjutnya menentukan *flowgraph* dari pengujian *flowchart* pengujian yang sudah dilakukan sebelumnya yang sesuai dengan *pseudocode* pada Tabel 4.23. Adapun berikut merupakan *flowgraph* algoritma metode *Probability Impact and Matrix* dapat dilihat pada Gambar 4.8.



Gambar 4.8 *Flowgraph* Algoritma Metode *Probability Impact and Matrix*

c. Tahap pengujian dalam menentukan *region*, *cyclomatic complexity*, *independent path*, *graph matrix*, dan *predicate node*. Berikut merupakan penjabarannya.

1) Region adalah area yang dibatasi oleh *edge* dan node disebut region, area diluar *graph* juga dihitung sebagai region. Rumus jumlah region adalah jumlah kurva tertutup ditambah 1 atau region terluar pada *flowgraph* yang sesuai pada *flowgraph* Gambar 4.8.

Jumlah region = 1 + 1 = 2

Terdapat tiga region dalam *flowgraph* pada Gambar 4.8.

2) Kompleksitas siklomatis adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Rumus kompleksitas siklomatis (*cyclomatic complexity*) adalah V(G) = E – N + 2. Dimana E adalah jumlah busur pada *flowgraph*, sedangkan N merupakan jumlah simpul pada *flowgraph* Gambar 4.8.

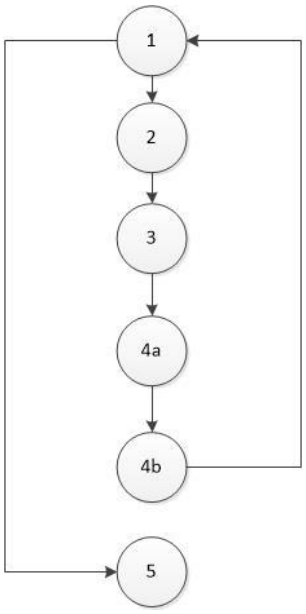
   V(G) = 6 – 6 + 2 = 2

3) Jalur *independent* adalah jalur yang terdapat dalam program yang memperkenalkan setidaknya satu rangkaian pernyataan proses atau kondisi baru. Sebuah *independent path* harus bergerak setidaknya sepanjang satu sisi yang belum ditelusuri sebelum jalur tersebut didefinisikan. Berdasarkan *flowgraph* pada Gambar 4.8 didapatkan jalur *independent* yang ada sebagai berikut:

   Path 1 : 1-6

   Path 2 : 1-2-3-4-5-1-6

   Selanjutnya penomoran ulang untuk *flowgraph* sesuai *flowgraph* sebelumnya pada Gambar 4.8. Adapun berikut merupakan *flowgraph* algoritma *Probability Impact and Matrix* dapat dilihat pada Gambar 4.9.



Gambar 4.9 *Flowgraph* Penomoran Ulang Algoritma PIM

4) *Graph matrix* adalah matrik berbentuk segiempat sama sisi, dimana jumlah baris dan kolom sama dengan jumlah node, dan identifikasi baris dan kolom sama dengan identifikasi node, serta isi data adalah keberadaan penghubung antar node (*edges*). Berdasarkan penomoran ulang pada *flowgraph* pada Gambar 4.9, adapun berikut *graph matrix* pada algoritma *probability impact and matrix* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Graph Matrix *Algoritma Metode Probability Impact Matrix*

	1	**2**	**3**	**4a**	**4b**	**5**	**n(E-1)**
1		1				1	1
2			1				0
3				1			0
4a					1		0
4b	1						0
5							0
						Sum(E)+1	1+1=2

5) *Predicate node* adalah proses yang membentuk dua cabang atau lebih atau node yang mempunyai kondisi berdasarkan *flowgraph* Gambar 4.8. *Predicate node* memiliki rumus yaitu V(G) = Jumlah *predicate node* + 1.

V(G) = 1 + 1 = 2

Predicate node pada algoritma *probability impact and matrix* adalah 2.

4.2.1.2 Kesimpulan Hasil Pengujian *White Box*

Berdasarkan hasil pengujian *white box* yang dilakukan dengan kasus uji tersebut, dapat ditarik kesimpulan sebagai berikut pada Tabel 4.25.

Tabel 4.25 Kesimpulan Pengujian *White Box*

No	**Kasus Uji**	**Hasil yang Diharapkan**	**Hasil Sesuai Uji Kasus**	**Keterangan**
1	Uji algoritma menggunakan *Critical Path Method*	Kondisi kegiatan pendahulu yang dipilih pengguna dapat membentuk solusi jawaban dalam menentukan fokus pengerjaan proyek dan membentuk sebuah solusi yang diharapkan	Kondisi kegiatan pendahulu yang dipilih pengguna dapat membentuk solusi jawaban dalam menentukan fokus pengerjaan proyek dan membentuk sebuah solusi yang diharapkan	[√] Alur Terlewati [ ] Alur Tidak Terlewati

No	Kasus Uji	Hasil yang Diharapkan	Hasil Sesuai Uji Kasus	Keterangan
2	Uji algoritma menggunakan metode *Earned Value Management*	Kondisi ketika melihat kemajuan pengerjaan proyek dengan benar berdasarkan laporan yang telah dikerjakan secara sempurna atau lengkap	Kondisi ketika melihat kemajuan pengerjaan proyek dengan benar berdasarkan laporan yang telah dikerjakan secara sempurna atau lengkap	[√] Alur Terlewati [ ] Alur Tidak Terlewati
3	Uji algoritma menggunakan metode *Probability Impact and Matrix*	Kondisi ketika pengguna melakukan manajemen resiko sehingga menghasilkan hasil level resiko berdasarkan dampak dan kemungkinan yang dipilih pengguna	Kondisi ketika pengguna melakukan manajemen resiko sehingga menghasilkan hasil level resiko berdasarkan dampak dan kemungkinan yang dipilih pengguna	[√] Alur Terlewati [ ] Alur Tidak Terlewati

Setelah melakukan pengujian algoritma metode manajemen proyek dengan pengujian *white box* maka dapat disimpulkan bahwa sebagai berikut:

1. Menjamin seluruh *independent path* di dalam modul yang dikerjakan sekurang-kurangnya sebanyak satu kali.
2. Mengerjakan seluruh keputusan logika dari sisi benar maupun salah.
3. Mengerjakan seluruh perulangan sesuai batasannya dan dalam batasan operasionalnya.
4. Mengerjakan seluruh struktur data internal yang menjamin validitas.

4.2.2 Skenario Pengujian *Black Box*

Berdasarkan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem manajemen proyek di PT Imeco Inter Sarana telah melalui tahap perbaikan dan sudah dimaksimalkan terhadap proses-proses tersebut, serta secara fungsional sistem sudah dapat digunakan dan menghasilkan keluaran yang diharapkan. Skenario pengujian *black box* dapat dilihat pada Tabel 4.26.

Tabel 4.26 Skenario Pengujian *Black Box*

Fungsional Uji	Butir Uji	Jenis Pengujian
Login	*Login* user	*Black Box*
Pengelolaan User	Menambah data user	*Black Box*
	Mengubah data user	
	Menghapus data user	
	Mencari data user	
Pengelolaan Master Material	Menambah data material	*Black Box*
	Mengubah data material	
	Menghapus data material	
	Mencari data material	
Pengelolaan Master Identifikasi	Menambah data identifikasi resiko	*Black Box*
	Mengubah data identifikasi resiko	
	Menghapus data identifikasi resiko	
	Mencari data identifikasi resiko	
Pengelolaan Master Pekerjaan	Menambah data pekerjaan	*Black Box*
	Mengubah data pekerjaan	
	Menghapus data pekerjaan	
	Mencari data pekerjaan	
Pengelolaan Klien	Menambah data klien	*Black Box*
	Mengubah data klien	
	Menghapus data klien	
	Mencari data klien	
Pengelolaan Proyek	Menambah data proyek	*Black Box*
	Mengubah data proyek	
	Menghapus data proyek	
	Mencari data proyek	
Membuat Laporan Pekerjaan	Menambah laporan kerja	*Black Box*
Menampilkan Laporan Pekerjaan	Melihat laporan pekerjaan supervisor	*Black Box*
Pengelolaan Struktur Kerja	Menambah data struktur kerja	*Black Box*
	Mengubah data struktur kerja	
	Menghapus data struktur kerja	
	Mencari data struktur kerja	
Pengelolaan Material	Menambah data material	*Black Box*
Pengelolaan Tingkat Kepentingan	Menambah data tingkat kepentingan	*Black Box*
	Mengubah data tingkat kepentingan	
	Menghapus data tingkat kepentingan	
	Mencari data tingkat kepentingan	
Pengelolaan Penanganan	Menambah data penanganan	*Black Box*
	Mengubah data penanganan	
	Menghapus data penanganan	
	Mencari data penanganan	

4.2.2.1 Kasus dan Hasil Pengujian *Black Box*

Berdasarkan rencana pengujian yang disusun, maka dilakukan pengujian sesuai dengan kelas uji sebelumnya. Berikut merupakan hasil pengujian *black box*.

1. Pengujian *Login*

Login dilakukan untuk penentuan hak akses pengguna admin, manajer, dan supervisor. Pengujian *login* untuk data yang benar dapat dilihat pada Tabel 4.27.

Tabel 4.27 Pengujian *Login* Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan *username* dan *password* dengan benar, contoh masukan, Username : manajer Password : 123456	*Field* data *username* dan *password* sudah terisi lalu menuju ke halaman utama sesuai hak akses pengguna	Menampilkan halaman utama sesuai hak akses pengguna	[√] Diterima [ ] Ditolak

Pengujian untuk *login* dengan data yang salah yang dilakukan pengguna dapat terlihat pada Tabel 4.28.

Tabel 4.28 Pengujian *Login* Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Field username atau *password* tidak diisi atau kosong, dan menekan tombol *Login*, contoh masukan, *Username* : *Password* :	Menampilkan pesan "Data tidak boleh kosong, harus diisi"	Menampilkan pesan "Data tidak boleh kosong, harus diisi"	[√] Diterima [ ] Ditolak
Memasukan data *username* dan *password* yang tidak terdaftar atau salah, contoh masukan: *Username* : manajer *Password* : admin	Menampilkan pesan "*Username* atau *password* yang anda masukkan salah"	Menampilkan pesan "*Username* atau *password* yang anda masukkan salah"	[√] Diterima [ ] Ditolak

2. Pengujian Pengelolaan User

Data user digunakan untuk penentuan hak akses pengguna. Pengujian pengelolaan data user dengan data benar dapat dilihat pada Tabel 4.29.

Tabel 4.29 Pengujian Pengelolaan User Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, Username : ancapur Nama Lengkap : Anca Purnama Password : 123456 Email : anca@gmail.com Status Akun : Aktif Status User : supervisor	Sistem menyimpan data dan menampilkan pesan "Sukses! Data user telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data user telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Username : ancapur Password lama : 123456 Password Baru : 654321 Konfirmasi Password Baru : 654321	Sistem mengubah data pada user yang dipilih dan menampilkan pesan "Sukses! Data user telah diubah"	Sistem mengubah data pada user yang dipilih dan menampilkan pesan "Sukses! Data user telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data user	Menampilkan pesan "Yakin menghapus data : Anca Purnama?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	Menampilkan pesan "Yakin menghapus data : Anca Purnama?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan dicari, Cari : Anca	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengelolaan data user dengan data salah dapat dilihat pada Tabel 4.30.

Tabel 4.30 Pengujian Pengelolaan User Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah user tidak diisi. Nama Lengkap : Inisial : Email : Username : Password : Status Akun : Status User :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* ubah user salah, contoh masukan Nama Lengkap : Admin Inisial : adm Email : admin @gmail.com Username:adm Password : 123 Status Akun : aktif Status User : admin	Tetap di halaman tambah user dan menampilkan pesan Username "harus 5-20 karakter" dan password "harus 6-20 karakter"	Tetap di halaman tambah user dan menampilkan pesan Username "harus 5-20 karakter" dan password "harus 6-20 karakter"	[√] Diterima [ ] Ditolak
Menghapus data user (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus data : Anca Purnama?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman user	Menampilkan pesan konfirmasi "Yakin menghapus data : Anca Purnama?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman user	[√] Diterima [ ] Ditolak
Data pada *form* cari user tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

3. Pengujian Pengelolaan Master Material

Data master material digunakan untuk pengelolaan material yang diperlukan pada pelaksanaan proyek. Pengujian pengelolaan data master material dengan data yang benar dapat dilihat pada Tabel 4.31.

Tabel 4.31 Pengujian Pengelolaan Master Material Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, Material : Tang L Satuan : Unit Harga : 35000	Sistem menyimpan data dan menampilkan pesan "Sukses! Data material telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data material telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Material : Tang L Satuan : Unit Harga : 40000	Sistem mengubah data pada material yang dipilih dan menampilkan pesan "Sukses! Data material telah diubah"	Sistem mengubah data pada material yang dipilih dan menampilkan pesan "Sukses! Data material telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data material	Menampilkan pesan "Yakin menghapus data : Tang L?", dan menampilkan pesan "Sukses! Data telah	Menampilkan pesan "Yakin menghapus data : Tang L?", dan menampilkan pesan "Sukses! Data telah	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
	dihapus" setelah pengguna memilih "ok"	dihapus" setelah pengguna memilih "ok"	
Contoh masukan yang akan dicari, Cari : Tang	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan data master material dengan data yang salah dapat dilihat pada Tabel 4.32.

Tabel 4.32 Pengujian Pengelolaan Master Material Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah material tidak diisi, contoh masukan, Material : Satuan : Harga :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak
Data pada *form* ubah material kosong, contoh masukan, Material : Satuan : Harga :	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Menghapus data material (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus material : Tang L?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman master material	Menampilkan pesan konfirmasi "Yakin menghapus material : Tang L?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman master material	[√] Diterima [ ] Ditolak
Data pada *form* cari material tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

4. Pengujian Pengelolaan Master Identifikasi

Data master identifikasi digunakan untuk pengelolaan identifikasi yang mungkin terjadi pada pelaksanaan proyek. Pengujian pengelolaan data identifikasi dengan data yang benar dapat dilihat pada Tabel 4.33.

Tabel 4.33 Pengujian Pengelolaan Master Identifikasi Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, Kode Resiko : R22 Jenis Resiko : Penggambaran Deskripsi : Salah spesifikasi	Sistem menyimpan data dan menampilkan pesan "Sukses! Data identifikasi telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data identifikasi telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Kode Resiko : R22 Jenis Resiko : Keuangan Deskripsi : Salah spesifikasi	Sistem mengubah data pada identifikasi yang dipilih dan menampilkan pesan "Sukses! Data identifikasi telah diubah"	Sistem mengubah data pada identifikasi yang dipilih dan menampilkan pesan "Sukses! Data identifikasi telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data identifikasi	Menampilkan pesan "Yakin menghapus resiko : Salah spesifikasi?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	Menampilkan pesan "Yakin menghapus resiko : Salah spesifikasi?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan dicari, Cari : Spesifikasi	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan data master identifikasi dengan data yang salah dapat dilihat pada Tabel 4.34.

Tabel 4.34 Pengujian Pengelolaan Master Identifikasi Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah identifikasi tidak diisi, contoh masukan, Kode Resiko : Jenis Resiko : Deskripsi :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak
Data pada *form* ubah identifikasi kosong, contoh masukan, Kode Resiko : Jenis Resiko : Deskripsi :	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Menghapus data identifikasi (*cancel*)	Menampilkan pesan konfirmasi "Yakin	Menampilkan pesan konfirmasi "Yakin	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
	menghapus resiko : Salah spesifikasi?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman master identifikasi	menghapus resiko : Salah spesifikasi?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman master identifikasi	
Data pada *form* cari identifikasi tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

5. Pengujian Pengelolaan Master Pekerjaan

Data master pekerjaan digunakan untuk pengelolaan kegiataan pada pelaksanaan proyek. Pengujian pengelolaan data pekerjaan dengan data yang benar dapat dilihat pada Tabel 4.35.

Tabel 4.35 Pengujian Pengelolaan Master Pekerjaan Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, Pekerjaan : Persiapan	Sistem menyimpan data dan menampilkan pesan "Sukses! Data pekerjaan telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data pekerjaan telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Pekerjaan : Penggambaran	Sistem mengubah data pada pekerjaan yang dipilih dan menampilkan pesan "Sukses! Data pekerjaan telah diubah"	Sistem mengubah data pada pekerjaan yang dipilih dan menampilkan pesan "Sukses! Data pekerjaan telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data pekerjaan	Menampilkan pesan "Yakin menghapus kegiatan : Persiapan?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	Menampilkan pesan "Yakin menghapus kegiatan : Persiapan?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan dicari, Cari : Persiapan	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan data master pekerjaan dengan data yang salah dapat dilihat pada Tabel 4.36.

Tabel 4.36 Pengujian Pengelolaan Master Pekerjaan Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah pekerjaan tidak diisi, contoh masukan, Pekerjaan :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak
Data pada *form* ubah pekerjaan kosong, contoh masukan, Pekerjaan :	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Menghapus data pekerjaan (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus kegiatan : Persiapan?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman master pekerjaan	Menampilkan pesan konfirmasi "Yakin menghapus kegiatan : Persiapan?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman master pekerjaan	[√] Diterima [ ] Ditolak
Data pada *form* cari pekerjaan tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

6. Pengujian Pengelolaan Klien

Data klien digunakan untuk pengelolaan konsumen terhadap proyek yang dikerjakan. Pengujian pengelolaan data klien dengan data yang benar dapat dilihat pada Tabel 4.37.

Tabel 4.37 Pengujian Pengelolaan Klien Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, SPK : 005-IME-PU Nama Lengkap : Widya L Telepon : 08126661902 Instansi : PT Pertamina	Sistem menyimpan data dan menampilkan pesan "Sukses! Data klien telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data klien telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, SPK : 005-IME-PU Nama Lengkap : Widya L Telepon : 085365619794 Instansi : PT Pertamina	Sistem mengubah data pada klien yang dipilih dan menampilkan pesan "Sukses! Data klien telah diubah"	Sistem mengubah data pada klien yang dipilih dan menampilkan pesan "Sukses! Data klien telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data klien	Menampilkan pesan "Yakin menghapus data klien?", dan menampilkan pesan "Sukses! Data telah	Menampilkan pesan "Yakin menghapus data klien?", dan menampilkan pesan "Sukses! Data telah	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
	dihapus" setelah pengguna memilih "ok"	dihapus" setelah pengguna memilih "ok"	
Contoh masukan yang akan dicari, Cari : Widya	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan data klien dengan data yang salah dapat dilihat pada Tabel 4.38.

Tabel 4.38 Pengujian Pengelolaan Klien Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah klien tidak diisi, contoh masukan, SPK : Nama Lengkap : Telepon : Instansi :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak
Data pada *form* ubah klien, contoh masukan, SPK : 005-IME-PU Nama Lengkap : Widya L Telepon : Abcdefghi Instansi : PT Pertamina	Menampilkan pesan "*Form* harus diisi angka"	Menampilkan pesan "*Form* harus diisi angka"	[√] Diterima [ ] Ditolak
Menghapus data klien (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus data klien?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman klien	Menampilkan pesan konfirmasi "Yakin menghapus data klien?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman klien	[√] Diterima [ ] Ditolak
Data pada *form* cari klien tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

7. Pengujian Pengelolaan Proyek

Data proyek digunakan untuk menampilkan data proyek yang ada pada perusahaan untuk dilakukan manajemen proyek. Pengujian pengelolaan data proyek dengan data yang benar dapat dilihat pada Tabel 4.39.

Tabel 4.39 Pengujian Pengelolaan Proyek Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, Kode Proyek : C228-142-006 Klien : 007/IME-PU/II/2016 | Ahmad Nama Proyek : PU Chevron Biaya : 1500000000 Status Proyek : Proses	Sistem menyimpan data dan menampilkan pesan "Sukses! Data proyek telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data proyek telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Kode Proyek : C228-142-006 Klien : 007/IME-PU/II/2016 | Ahmad Nama Proyek : PU Chevron Biaya : 1500000000 Status Proyek : Selesai	Sistem mengubah data pada proyek yang dipilih dan menampilkan pesan "Sukses! Data proyek telah diubah"	Sistem mengubah data pada proyek yang dipilih dan menampilkan pesan "Sukses! Data proyek telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data proyek	Menampilkan pesan "Yakin menghapus proyek?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	Menampilkan pesan "Yakin menghapus data proyek?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan dicari, Cari : Pertamina	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan data proyek dengan data yang salah dapat dilihat pada Tabel 4.40.

Tabel 4.40 Pengujian Pengelolaan Proyek Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah proyek tidak diisi, contoh masukan, Kode Proyek : Klien : Nama Proyek : Biaya :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Status Proyek :			
Data pada *form* ubah proyek kosong, contoh masukan, Kode Proyek : Klien : Nama Proyek : Biaya : Status Proyek :	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	Menampilkan pesan "*Form* tidak boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Menghapus data proyek (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus data proyek?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman proyek	Menampilkan pesan konfirmasi "Yakin menghapus data proyek?" Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman proyek	[√] Diterima [ ] Ditolak
Data pada *form* cari proyek tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

8. Pengujian Membuat Laporan Pekerjaan

Data membuat laporan pekerjaan digunakan untuk pelaporan yang dimasukkan oleh supervisor terhadap pekerjaan yang dikerjakan. Pengujian membuat laporan pekerjaan dengan data yang benar dapat dilihat pada Tabel 4.41.

Tabel 4.41 Pengujian Membuat Laporan Pekerjaan Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data, Laporan Pekerjaan : Pengerjaan penggambaran PU telah selesai dilakukan Status : Selesai	Sistem menyimpan data dan menampilkan pesan "Data laporan telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Data laporan telah disimpan"	[√] Diterima [ ] Ditolak

Pengujian untuk membuat laporan kerja dengan data yang salah dapat dilihat pada Tabel 4.42.

Tabel 4.42 Pengujian Membuat Laporan Kerja Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah laporan tidak diisi, contoh	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
masukan, Laporan Pekerjaan : Status : Selesai	diisi"	diisi"	

9. Pengujian Menampilkan Laporan Pekerjaan

Data menampilkan laporan pekerjaan digunakan untuk menampilkan laporan yang dimasukkan oleh supervisor dan menampilkan jadwal kerja proyek. Pengujian menampilkan laporan pekerjaan dengan data yang benar dapat dilihat pada Tabel 4.43.

Tabel 4.43 Pengujian Menampilkan Laporan Pekerjaan Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik nama supervisor pada setiap jadwal berdasarkan kegiatan yang dilakukan	Sistem menampilkan data yang dimasukkan oleh supervisor	Sistem menampilkan data yang dimasukkan oleh supervisor	[√] Diterima [ ] Ditolak

Pengujian untuk menampilkan laporan kerja dengan data yang salah dapat dilihat pada Tabel 4.44.

Tabel 4.44 Pengujian Menampilkan Laporan Kerja Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik pada selain nama supervisor pada setiap jadwal berdasarkan kegiatan yang dilakukan	Sistem tidak menampilkan data yang dimasukkan oleh supervisor karena tidak ada kegiatan yang dilakukan	Sistem tidak menampilkan data yang dimasukkan oleh supervisor karena tidak ada kegiatan yang dilakukan	[√] Diterima [ ] Ditolak

10. Pengujian Pengelolaan Struktur Kerja

Data struktur kerja digunakan untuk pengelolaan struktur kerja kegiatan proyek. Pengujian pengelolaan data struktur kerja dengan data yang benar dapat dilihat pada Tabel 4.45.

Tabel 4.45 Pengujian Pengelolaan Struktur Kerja Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data struktur kerja "persiapan", Nama Kegiatan : Penyediaan sumber	Sistem menyimpan data dan menampilkan pesan "Sukses! Data struktur kerja telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data struktur kerja telah disimpan"	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
daya Tanggal Pengerjaan : 11/01/2016 s/d 13/01/2016 Biaya : 2500000 Supervisor : Anca Purnama			
Contoh masukan yang akan diubah, Nama Kegiatan : Penyediaan sumber daya Tanggal Pengerjaan : 11/01/2016 s/d 12/01/2016 Biaya : 1500000 Supervisor : Anca Purnama	Sistem mengubah data pada kegiatan struktur kerja yang dipilih dan menampilkan pesan "Sukses! Data struktur kerja telah diubah"	Sistem mengubah data pada kegiatan struktur kerja yang dipilih dan menampilkan pesan "Sukses! Data struktur kerja telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data struktur kerja	Menampilkan pesan "Yakin menghapus kegiatan : Penyediaan sumber daya?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	Menampilkan pesan "Yakin menghapus kegiatan : Penyediaan sumber daya?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan dicari, Cari : Persiapan	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan struktur kerja dengan data yang salah dapat dilihat pada Tabel 4.46.

Tabel 4.46 Pengujian Pengelolaan Struktur Kerja Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah struktur kerja tidak diisi, contoh masukan, Nama Kegiatan : Tanggal Pengerjaan : Biaya : Supervisor :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak
Data pada *form* ubah struktur kerja, contoh masukan, Nama Kegiatan : Penyediaan sumber	Menampilkan pesan "*Form* harus diisi benar"	Menampilkan pesan "*Form* harus diisi benar"	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
daya Tanggal Pengerjaan : 11/01/2016 s/d 12/01/2016 Biaya : Abcdefgh Supervisor : Anca Purnama			
Menghapus data struktur kerja (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus kegiatan : Penyediaan sumber daya?", Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman struktur kerja	Menampilkan pesan konfirmasi "Yakin menghapus kegiatan : Penyediaan sumber daya?", Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman struktur kerja	[√] Diterima [ ] Ditolak
Data pada *form* cari struktur kerja tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

11. Pengujian Data Material

Data material digunakan untuk pengelolaan material pada setiap kegiatan yang ada pada struktur kerja. Pengujian data material dengan data yang benar dapat dilihat pada Tabel 4.47.

Tabel 4.47 Pengujian Material Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data material, Material : Kunci L Jumlah : 2	Sistem menyimpan data dan menampilkan pesan "Sukses! Data material telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data material telah disimpan"	[√] Diterima [ ] Ditolak

Pengujian untuk data material dengan data yang salah dapat dilihat pada Tabel 4.48.

Tabel 4.48 Pengujian Material Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah material tidak diisi, contoh masukan, Material : Kunci L Jumlah :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak

12. Pengujian Pengelolaan Tingkat Kepentingan

Data tingkat kepentingan digunakan untuk pengelolaan tingkat kepentingan resiko pada proyek. Pengujian pengelolaan tingkat kepentingan dengan data yang benar dapat dilihat pada Tabel 4.49.

Tabel 4.49 Pengujian Pengelolaan Tingkat Kepentingan Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data tingkat kepentingan, Identifikasi : Spesifikasi gambar tidak lengkap Kemungkinan : 1 Dampak : 2 Tingkat Kepentingan : 2	Sistem menyimpan data dan menampilkan pesan "Sukses! Data tingkat kepentingan telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data tingkat kepentingan telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Identifikasi : Spesifikasi gambar tidak lengkap Kemungkinan : 1 Dampak : 3 Tingkat Kepentingan : 3	Sistem mengubah data pada kegiatan tingkat kepentingan yang dipilih dan menampilkan pesan "Sukses! Data tingkat kepentingan telah diubah"	Sistem mengubah data pada kegiatan tingkat kepentingan yang dipilih dan menampilkan pesan "Sukses! Data tingkat kepentingan telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data tingkat kepentingan	Menampilkan pesan "Yakin menghapus resiko : Spesifikasi gambar tidak lengkap?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	Menampilkan pesan "Yakin menghapus resiko : Spesifikasi gambar tidak lengkap?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan dicari, Cari : Spesifikasi	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan tingkat kepentingan dengan data yang salah dapat dilihat pada Tabel 4.50.

Tabel 4.50 Pengujian Pengelolaan Tingkat Kepentingan Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah tingkat	Menampilkan pesan "*Form* tidak boleh ada	Menampilkan pesan "*Form* tidak boleh ada	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
kepentingan tidak diisi, contoh masukan, Identifikasi : Kemungkinan : Dampak : Tingkat Kepentingan :	yang kosong, harus diisi"	yang kosong, harus diisi"	
Menghapus data tingkat kepentingan (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus resiko : Spesifikasi gambar tidak lengkap?", Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman tingkat kepentingan	Menampilkan pesan konfirmasi"Yakin menghapus resiko : Spesifikasi gambar tidak lengkap?", Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman tingkat kepentingan	[√] Diterima [ ] Ditolak
Data pada *form* cari tingkat kepentingan tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

13. Pengujian Pengelolaan Penanganan

Data penangan digunakan untuk pengelolaan penangan resiko berdasarkan identifikasi resiko pada pelaksanaan proyek. Pengujian pengelolaan data penanganan dengan data yang benar dapat dilihat pada Tabel 4.51.

Tabel 4.51 Pengujian Pengelolaan Penanganan Data Benar

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh masukan tambah data penanganan, Tambah Tindakan : Membuat spesifikasi dengan klien secara rinci	Sistem menyimpan data dan menampilkan pesan "Sukses! Data penanganan telah disimpan"	Sistem menyimpan data dan menampilkan pesan "Sukses! Data penanganan telah disimpan"	[√] Diterima [ ] Ditolak
Contoh masukan yang akan diubah, Edit Tindakan : Membuat spesifikasi dengan klien secara rinci sebelum pekerjaan dilaksanakan	Sistem mengubah data pada penanganan yang dipilih dan menampilkan pesan "Sukses! Data penanganan telah diubah"	Sistem mengubah data pada kegiatan struktur kerja yang dipilih dan menampilkan pesan "Sukses! Data penanganan telah diubah"	[√] Diterima [ ] Ditolak
Menghapus data penanganan	Menampilkan pesan "Yakin menghapus data : Membuat	Menampilkan pesan "Yakin menghapus data : Membuat	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
	spesifikasi dengan klien secara rinci sebelum pekerjaan dilaksanakan?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	spesifikasi dengan klien secara rinci sebelum pekerjaan dilaksanakan?", dan menampilkan pesan "Sukses! Data telah dihapus" setelah pengguna memilih "ok"	
Contoh masukan yang akan dicari, Cari : Spesifikasi	Sistem menapilkan hasil pencarian berdasarkan kata kunci	Sistem menapilkan hasil pencarian berdasarkan kata kunci	[√] Diterima [ ] Ditolak

Pengujian untuk pengolahan penanganan dengan data yang salah dapat dilihat pada Tabel 4.52.

Tabel 4.52 Pengujian Pengelolaan Penanganan Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data pada *form* tambah penanganan tidak diisi, contoh masukan, Tambah Tindakan :	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	Menampilkan pesan "*Form* tidak boleh ada yang kosong, harus diisi"	[√] Diterima [ ] Ditolak
Data pada *form* ubah penanganan, contoh masukan, Edit Tindakan :	Menampilkan pesan "*Form* harus diisi benar"	Menampilkan pesan "*Form* harus diisi benar"	[√] Diterima [ ] Ditolak
Menghapus data penanganan (*cancel*)	Menampilkan pesan konfirmasi "Yakin menghapus data : Membuat spesifikasi dengan klien secara rinci sebelum pekerjaan dilaksanakan?", Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman penanganan	Menampilkan pesan konfirmasi "Yakin menghapus data : Membuat spesifikasi dengan klien secara rinci sebelum pekerjaan dilaksanakan?", Jika tekan *cancel*, maka data tidak akan terhapus dan tetap di halaman penanganan	[√] Diterima [ ] Ditolak
Data pada *form* cari penanganan tidak ditemukan.	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	Menampilkan pesan "Hasil pencarian kosong atau tidak ditemukan"	[√] Diterima [ ] Ditolak

4.2.2.2 Kesimpulan Hasil Pengujian *Black Box*

Berdasarkan hasil pengujian *black box* yang dilakukan dengan kasus uji tersebut, dapat ditarik kesimpulan sebagai berikut:

1. Memberikan kesimpulan bahwa pada proses sudah benar saat memasukkan data benar.
2. Penyaringan kesalahan dalam bentuk tampilan pesan sehingga dapat membantu mengarahkan pengguna agar memberi masukan yang benar.
3. Secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan baik dalam pengelolaan pengguna, pengelolaan klien dan proyek, perencanaan jadwal proyek, biaya pengerjaan proyek, pengawasan pengerjaan proyek baik dalam mengidentifikasi level resiko dan pencapaian kemajuan proyek.

4.3 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan mengenai kepuasan pengguna dengan kandungan poin yaitu pemenuhan kebutuhan dari tujuan awal pembangunan sistem manajemen proyek di PT Imeco Inter Sarana dan tampilan antarmuka dari sistem manajemen proyek di PT Imeco Inter Sarana tersebut. Pengujian beta dilakukan melalui sebuah teknik pengambilan data, yaitu melalui wawancara.

4.3.1 Wawancara Pengguna

Wawancara dilakukan dengan menggunakan teknik kualitatif melalui pertanyan terbuka untuk pengguna dari sistem yang terdiri dari manajer proyek, supervisor, dan admin. Hasil wawancara tersebut akan dilakukan pengambilan kesimpulan terhadap penilaian penerapan sistem yang baru. Wawancara terdiri dari beberapa pertanyaan terbuka yang akan ditanyakan kepada narasumber.

4.3.1.1 Wawancara Pengujian Beta Manajer Proyek

Wawancara dilakukan untuk pengujian beta terhadap sistem manajemen proyek kepada Bapak Hermanu Sutanto selaku bagian *Project Manager* di perusahaan. Kegiatan wawancara ini dilakukan pada tanggal 12 Juli 2016 yang bertempat di kantor utama PT Imeco Inter Sarana. Adapun daftar pertanyaan dan hasil wawancara dapat dilihat pada Tabel 4.53.

Tabel 4.53 Wawancara Pengujian Beta Terhadap Manajer Proyek

No	Pertanyaan	Jawaban
1	Apakah sistem manajemen proyek ini mudah digunakan?	Bapak Hermanu Sutanto menyatakan bahwa aplikasi ini sederhana dan dapat dengan mudah untuk dipahami
2	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam perencanaan jadwal proyek?	Bapak Hermanu Sutanto menyatakan bahwa dengan aplikasi ini, perencanaan proyek untuk mendapatkan hasil perhitungan jadwal dan biaya mudah dimasukkan
3	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam pengawasan pengerjaan proyek?	Bapak Hermanu Sutanto menyatakan bahwa dengan aplikasi ini, pengawasan pengerjaan proyek mudah untuk dilaksanakan, terutama pada jadwal kerja dengan adanya *gantt chart* dan catatan dari supervisor pada tiap jadwal
4	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam mengidentifikasi dan penanganan resiko proyek?	Bapak Hermanu Sutanto menyatakan bahwa aplikasi ini mempermudah dalam mengurangi atau menghindari resiko yang mungkin akan terjadi pada pelaksanaan proyek dengan melihat bobot penilaian resiko dan penaganan
5	Bagaimana menurut anda tentang bahasa yang digunakan pada sistem manajemen proyek ini?	Bapak Hermanu Sutanto menyatakan bahwa bahasa pada sistem manajemen proyek ini mudah untuk digunakan, terutama untuk pengguna yang telah terbiasa menggunakan *website*
6	Bagaimana pendapat anda tentang tampilan antarmuka sistem manajemen proyek ini?	Bapak Hermanu Sutanto menyatakan bahwa tampilan antarmuka (*interface*) sistem manajemen ini terlihat agak kaku dan disarankan untuk menambah warna lain agar terlihat lebih menarik

Berdasarkan hasil wawancara dengan Bapak Hermanu Sutanto yang akan menjadi pengguna manajer. Dapat ditarik kesimpulan bahwa, sistem sudah baik dalam mengelola data klien, proyek, perencenaan kegiatan proyek, melihat manajemen proyek dan resiko, serta penggunaan dari bahasa maupun tampilan antarmuka mudah untuk dimengerti dengan sedikit arahan dalam penggunaannya.

4.3.1.2 Wawancara Pengujian Beta Supervisor

Wawancara dilakukan untuk pengujian beta terhadap sistem manajemen proyek kepada Bapak Risman selaku bagian *Workshop Supervisor* di perusahaan. Kegiatan wawancara ini dilakukan pada tanggal 12 Agustus 2016 yang bertempat di kantor utama PT Imeco Inter Sarana. Adapun daftar pertanyaan dan hasil wawancara dapat dilihat pada Tabel 4.54.

Tabel 4.54 Wawancara Pengujian Beta Terhadap Supervisor

No	Pertanyaan	Jawaban
1	Apakah sistem manajemen proyek ini mudah digunakan?	Bapak Risman menyatakan bahwa aplikasi ini cukup mudah digunakan
2	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam perencanaan jadwal proyek?	Bapak Risman menyatakan bahwa dengan aplikasi ini dapat mudah dalam menambahkan kegiatan pekerjaan, waktu, dan biaya
3	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam memberikan laporan pengerjaan proyek?	Bapak Risman menyatakan bahwa dengan aplikasi ini mudah dalam menambahkan catatan laporan pekerjaan sehingga dapat dengan mudah melaporkan kepada manajer
4	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam membuat data material proyek?	Bapak Risman menyatakan bahwa aplikasi ini mudah dalam penambahan material pada setiap kegiatan pekerjaan
5	Bagaimana menurut anda tentang bahasa yang digunakan pada sistem manajemen proyek ini?	Bapak Risman menyatakan bahwa bahasa pada sistem manajemen proyek ini cukup untuk dimengerti penggunaannya
6	Bagaimana pendapat anda tentang tampilan antarmuka sistem manajemen proyek ini?	Bapak Risman menyatakan bahwa tampilan antarmuka (*interface*) sistem manajemen ini cukup rapi.

Berdasarkan hasil wawancara dengan Bapak Risman yang akan menjadi pengguna supervisor. Dapat ditarik kesimpulan bahwa, sistem sudah baik dalam mengelola data struktur kegiatan proyek, melihat manajemen proyek, pencatatan laporan proyek, dan melihat kemajuan proyek, serta penggunaan dari bahasa maupun tampilan antarmuka mudah untuk dimengerti.

4.3.1.3 Wawancara Pengujian Beta Admin

Wawancara dilakukan untuk pengujian beta terhadap sistem manajemen proyek kepada Bapak Nirwan Yahya selaku bagian staf HRD atau admin di perusahaan. Kegiatan wawancara ini dilakukan pada tanggal 12 Agustus 2016 yang bertempat di kantor utama PT Imeco Inter Sarana. Adapun daftar pertanyaan dan hasil wawancara dapat dilihat pada Tabel 4.55.

Tabel 4.55 Wawancara Pengujian Beta Terhadap Admin

No	Pertanyaan	Jawaban
1	Apakah sistem manajemen proyek ini mudah digunakan?	Bapak Nirwan Yahya menyatakan bahwa aplikasi ini cukup mudah dalam penggunaannya
2	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam pengelolaan pengguna sistem?	Bapak Nirwan Yahya menyatakan bahwa dengan aplikasi ini sangat membantu dalam penambahan atau pengubahan dalam pengelolaan pengguna, sehinga mudah

No	Pertanyaan	Jawaban
		dalam pengaktifan akun pengguna pada sistem
3	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam pengelolaan data master material proyek?	Bapak Nirwan Yahya menyatakan bahwa dengan aplikasi ini sangat membantu dalam pengelolaan master material karena dapat dengan mudah dalam pengelolaan data material yang ada di gudang proyek
4	Apakah dengan adanya sistem manajemen proyek ini dapat membantu anda dalam pengelolaan data master identifikasi resiko proyek?	Bapak Nirwan Yahya menyatakan bahwa aplikasi ini mempermudah menambahkan pengelolaan identifikasi resiko yang mungkin akan terjadi pada saat pelaksanaan proyek
5	Bagaimana menurut anda tentang bahasa yang digunakan pada sistem manajemen proyek ini?	Bapak Nirwan Yahya menyatakan bahwa bahasa pada sistem manajemen proyek ini mudah untuk dimengerti dan sederhana
6	Bagaimana pendapat anda tentang tampilan antarmuka sistem manajemen proyek ini?	Bapak Nirwan Yahya menyatakan bahwa tampilan antarmuka (*interface*) sistem manajemen ini cukup bagus tetapi disarankan penambahan tampilan yang lebih menarik seperti fitur yang bergerak

Berdasarkan hasil wawancara dengan Bapak Nirwan Yahya yang akan menjadi pengguna admin. Dapat ditarik kesimpulan bahwa, sistem sudah baik dalam mengelola data pengguna, data material, serta data dalam pengidentifikasian resiko, serta penggunaan dari bahasa maupun tampilan antarmuka mudah untuk dimengerti dengan penambahan saran dalam fitur tampilan bergerak.

4.3.2 Kesimpulan Hasil Pengujian Beta

Berdasarkan hasil jawaban dari bagian manajer, supervisor, serta admin di PT Imeco Inter Sarana terhadap pertanyaan yang diajukan pada pengujian beta, maka dapat ditarik kesimpulan bahwa sistem yang dibangun dapat membantu perusahaan dalam melakukan proses perencanaan proyek baik dalam pengelolaan struktur kegiatan pekerjaan, jadwal, biaya, membantu perusahaan dalam pengawasan pengerjaan yang dilaksanakan oleh tenaga kerja dalam pengawasan laporan kerja serta melihat pencapaian kemajuan proyek, dan sistem manajemen proyek mudah digunakan dan mudah dimengerti, namun kurang menarik dari sisi tampilan.