BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

IV.1 Implementasi

Tahap implementasi merupakan kelanjutan dari tahap perancangan sistem. Tujuan implementasi adalah untuk menerapkan perancangan yang telah dilakukan terhadap sistem, sehingga user dapat memberi masukkan demi berkembangnya sistem yang telah dibangun menjadi lebih baik. Tahap implementasi dibagi menjadi empat bagian yaitu implementasi perangkat keras, perangkat lunak, basis data dan antarmuka.

IV.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras menjelaskan perangkat keras yang digunakan untuk mengimplementasikan sistem yang telah dibuat. Perangkat keras yang digunakan dalam implementasi sistem ini dapat dilihat pada tabel IV.1.

Tabel IV.1 Perangkat Keras yang digunakan

No.	Perangkat Keras	Spesifikasi
1	Processor	Kecepatan 3,0 GHz
2	Monitor	LCD 17 Inch
3	VGA	VGA On-board 128 MB
4	Memory	2 GB
5	Harddisk	160 GB
6	Printer	Printer Hitam Putih

IV.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan perangkat lunak yang digunakan dalam implementasi sistem ini. Perangkat lunak yang digunakan dapat dilihat pada tabel IV.Tabel IV.2 Perangkat Lunak yang digunakan

No	Perangkat lunak yang dibutuhkan
1.	Sistem informasi yang digunakan *Microsoft Windows XP Professional Service Pack 3*

IV.1.3 Implementasi Basis Data

Implementasi basis data menjelaskan implementasi dari basis data yang dibuat pada sistem ini, meliputi implementasi proses pembuatan *database* dan *table*. Implementasi basis data pada sistem ini dapat dilihat pada tabel IV.3.

Tabel IV.3 Implementasi Basis Data

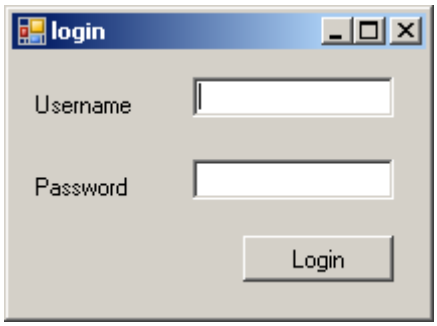
No.	Nama operaasi	Keterangan
1	Pembuatan database	CREATE DATABASE [pdskm] ON PRIMARY (NAME = N'pdskm', FILENAME = N'c:\Program Files (x86)\Microsoft SQL Server\MSSQL10_50.SQLRIZKI\MSSQL\DATA\pdskm.mdf' , SIZE = 3072KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB) LOG ON (NAME = N'pdskm_log', FILENAME = N'c:\Program Files (x86)\Microsoft SQL Server\MSSQL10_50.SQLRIZKI\MSSQL\DATA\pdskm.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
2	Pembuatan table bestek	CREATE TABLE [dbo].[bestek]([bestek_id] [int] IDENTITY(1,1) NOT NULL, [jenis_bestek_id] [int] NOT NULL, [panjang] [int] NOT NULL, [lebar] [int] NOT NULL, [tinggi] [int] NOT NULL, [aktif] [bit] NOT NULL, CONSTRAINT [PK_bestek] PRIMARY KEY CLUSTERED ([bestek_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY] ALTER TABLE [dbo].[bestek] WITH CHECK ADD CONSTRAINT [FK_bestek_jenis_bestek] FOREIGN KEY([jenis_bestek_id]) REFERENCES [dbo].[jenis_bestek] ([jenis_bestek_id])
3	Pembuatan table jenisBestek	CREATE TABLE [dbo].[jenis_bestek]([jenis_bestek_id] [int] IDENTITY(1,1) NOT NULL, [jenis] [varchar](50) NOT NULL, [aktifJenis] [bit] NOT NULL, CONSTRAINT [PK_jenis_bestek] PRIMARY KEY CLUSTERED ([jenis_bestek_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
4	Pembuatan table pemenuhan pemesanan	CREATE TABLE [dbo].[pemenuhan_pemesanan]([pemenuhan_pesanan_id] [int] IDENTITY(1,1) NOT NULL, [pemesanan_detail_id] [int] NOT NULL, [stok_id] [int] NOT NULL,

		[jumlah] [int] NOT NULL, [proses] [int] NOT NULL, [sisa] [int] NOT NULL, CONSTRAINT [PK_pemenuhan_pemesanan_1] PRIMARY KEY CLUSTERED ([pemenuhan_pesanan_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY] ALTER TABLE [dbo].[pemenuhan_pemesanan] WITH CHECK ADD CONSTRAINT [FK_pemenuhan_pemesanan_pemesanan_detail] FOREIGN KEY([pemesanan_detail_id]) REFERENCES [dbo].[pemesanan_detail] ([pemesanan_detail_id])
5	Pembuatan table pemesanan	CREATE TABLE [dbo].[pemesanan]([pemesanan_id] [int] IDENTITY(1,1) NOT NULL, [nama_pemesan] [varchar](50) NOT NULL, [alamat_pemesan] [varchar](max) NOT NULL, [no_telepon_pemesan] [varchar](50) NOT NULL, [tgl_pesan] [datetime] NOT NULL, [tgl_jadi] [datetime] NULL, [status] [int] NOT NULL, CONSTRAINT [PK_pemesanan] PRIMARY KEY CLUSTERED ([pemesanan_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
6	Pembuatan table pemesanan _detail	CREATE TABLE [dbo].[pemesanan_detail]([pemesanan_detail_id] [int] IDENTITY(1,1) NOT NULL, [pemesanan_id] [int] NOT NULL, [bestek_id] [int] NOT NULL, [jumlah] [int] NOT NULL, CONSTRAINT [PK_pemesanan_detail] PRIMARY KEY CLUSTERED ([pemesanan_detail_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY] ALTER TABLE [dbo].[pemesanan_detail] WITH CHECK ADD CONSTRAINT [FK_pemesanan_detail_pemesanan] FOREIGN KEY([pemesanan_id]) REFERENCES [dbo].[pemesanan] ([pemesanan_id])
7	Pembuatan table stok_kayu	CREATE TABLE [dbo].[stok_kayu]([stok_id] [int] IDENTITY(1,1) NOT NULL, [diameter] [int] NOT NULL, [tinggi] [int] NOT NULL, [jumlah] [int] NOT NULL, [aktif] [bit] NOT NULL, CONSTRAINT [PK_stok_kayu] PRIMARY KEY CLUSTERED

		([stok_id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
8	Pembuatan table user	CREATE TABLE [dbo].[user]([user_id] [int] NOT NULL, [username] [varchar](50) NOT NULL, [password] [varchar](max) NOT NULL) ON [PRIMARY]

IV.1.4 Implementasi Antarmuka

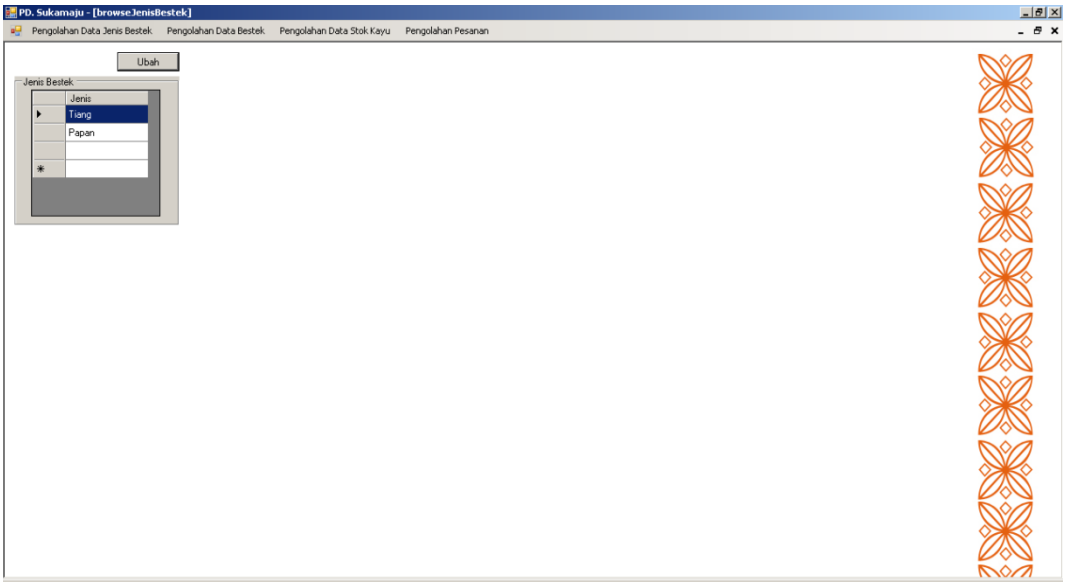
Implementasi antarmuka merupakan tampiln dari aplikasi yang dibangun. Implementasi pada sistem yang dibangun dapat dilihat pada gambar IV.1 sampai IV.14.



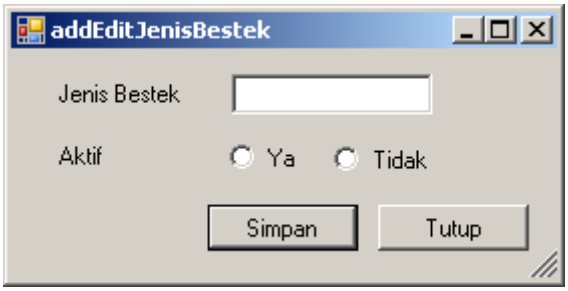
Gambar IV.1 Login



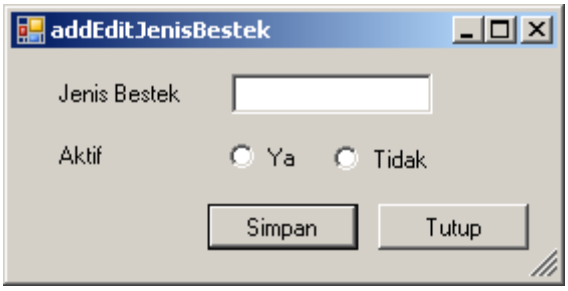
Gambar IV.2 Form Utama



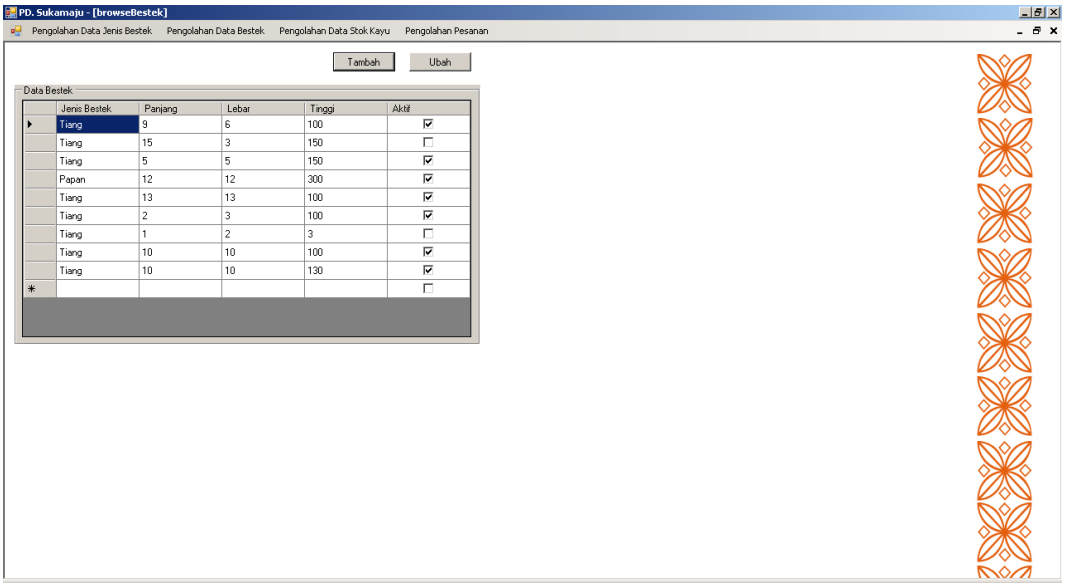
Gambar IV.3 Daftar Jenis Bestek



Gambar IV.4 Tambah Jenis Bestek



Gambar IV.5 Edit Jenis Bestek



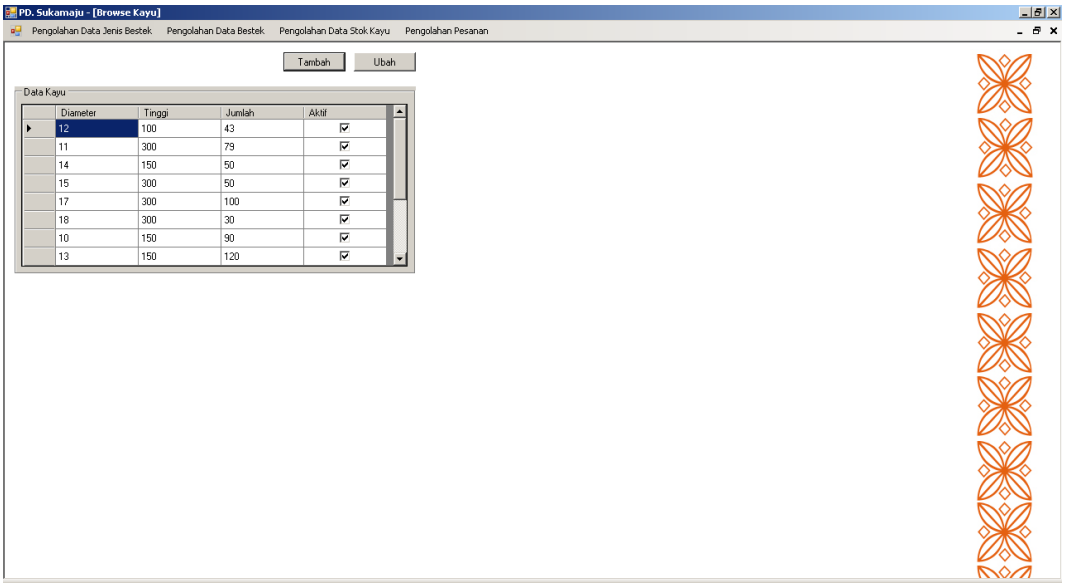
Gambar IV.6 Daftar Bestek



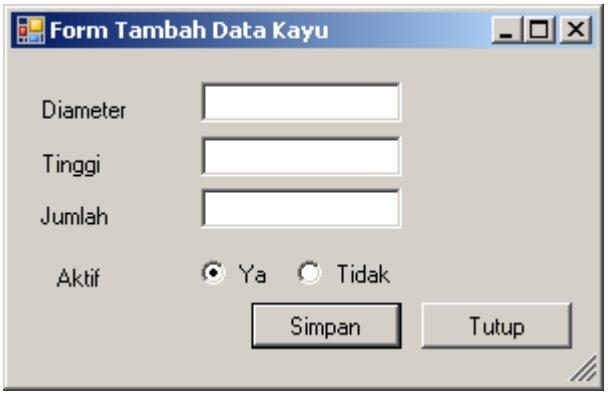
Gambar IV.7 Tambah Bestek



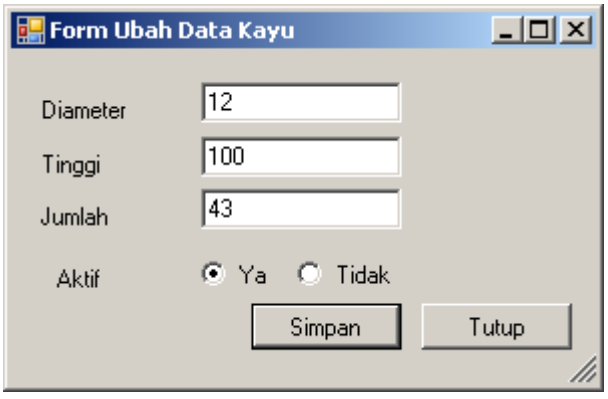
Gambar IV.8 Ubah Bestek



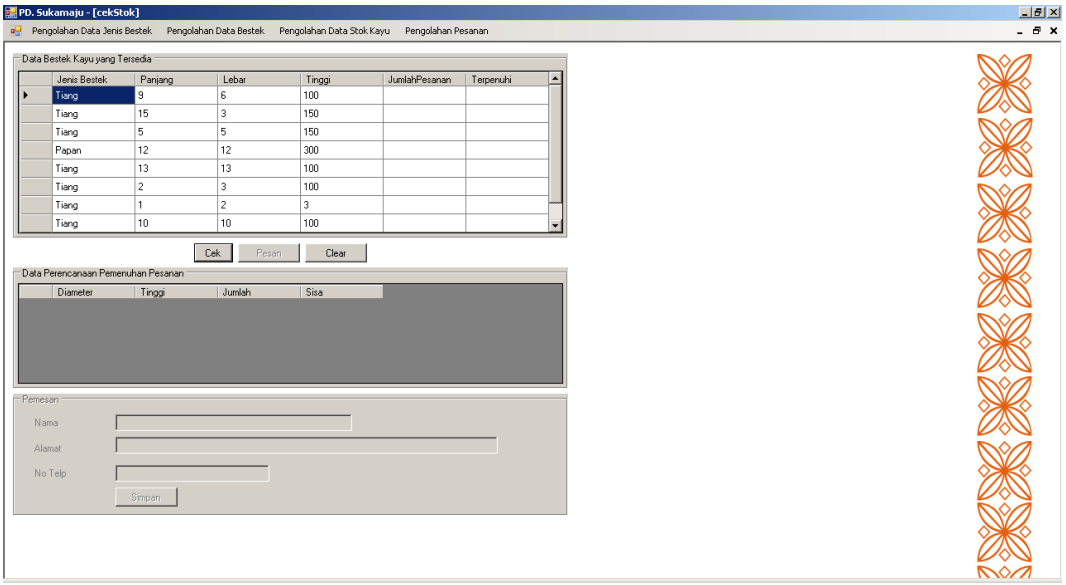
Gambar IV.9 Daftar Kayu



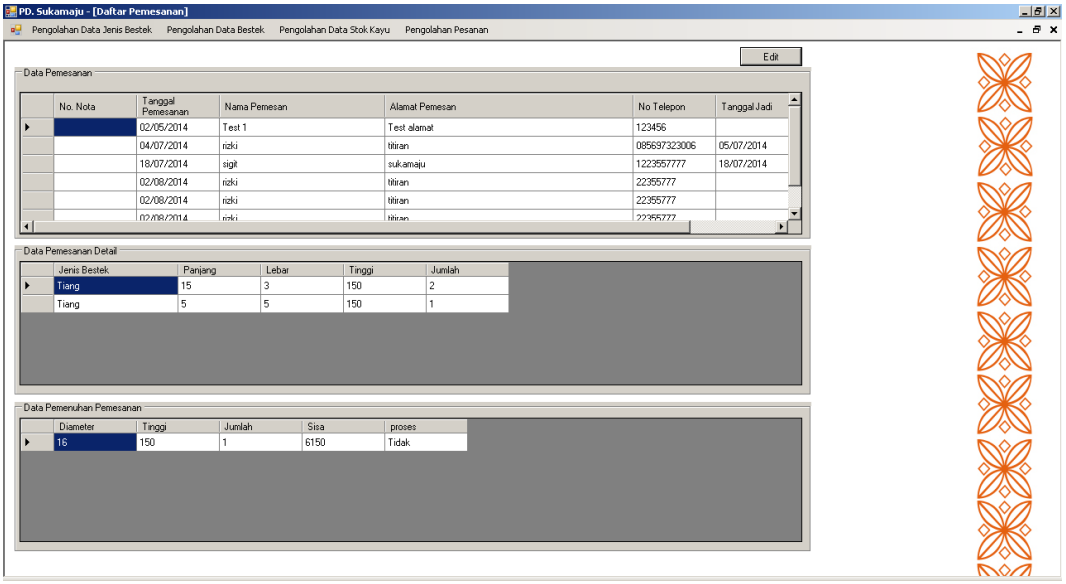
Gambar IV.10 tambah Data Kayu



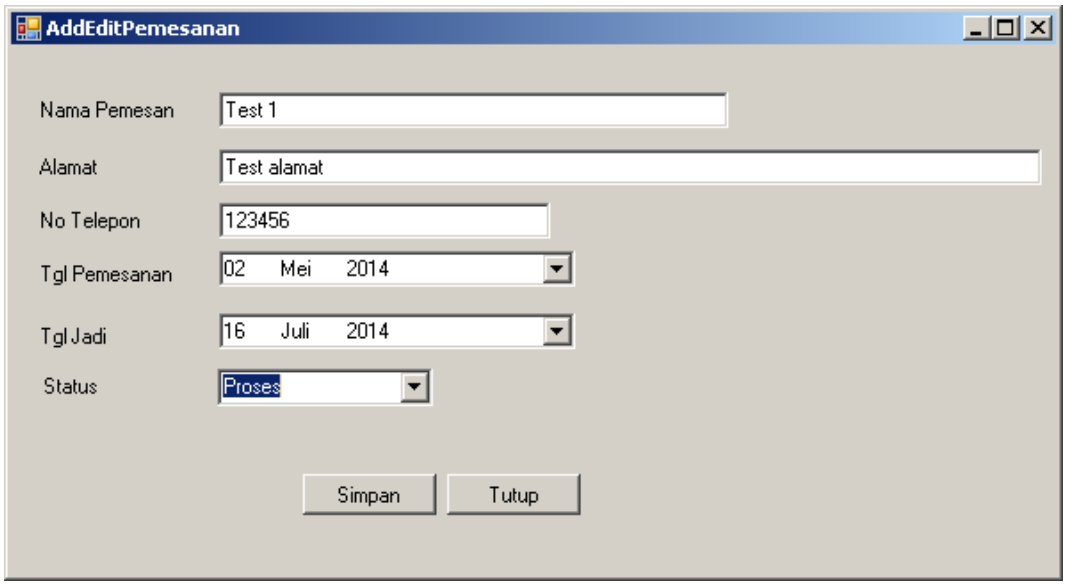
Gambar IV.11 Ubah Data Kayu



Gambar IV.12 Cek Stok



Gambar IV.13 Daftar Pemesanan



Gambar IV.14 Edit Pemesanan

IV.2 Pengujian Perangkat Lunak

Pengujian merupakan bagian penting dalam pengembangan suatu perangkat lunak. Tujuan dari pengujian adalah menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian dilakukan untuk mengetahui perangkat lunak kesesuaian aplikasi yang telah dibuat dengan perancangan aplikasi tersebut.

Pengujian yang dilakukan adalah pengujian *black box* dan white box. Pengujian *black box* digunakan untuk menguji fungsionalitas yang ada pada aplikasi. Pengujian white box digunakan untuk menguji sistem serta digunakan untuk meyakinkan semua perintah dan kondisi dalam proses eksekusi secara minimal. Pengujian *black box* dan white box yang ada pada aplikasi dapat dilihat pada tabel IV.4.

Tabel IV.4 Rencana Pengujian

No.	Kelas Uji	Butir Uji	Jenis Pengujian
1	Login	Masuk ke sistem	*Black box*
2	Menu Utama	Memilih Jenis Bestek	*Black box*
		Memilih Data Bestek	*Black box*
		Memilih Stok Kayu	*Black box*
		Memilih Pengecekkan Stok	*Black box*
		Memilih Daftar Pesanan	*Black box*
		Memilih Laporan Stok	*Black box*
		Memilih Laporan Pesanan	*Black box*
3	Jenis Bestek	Tambah Data Jenis Bestek	*Black box*

		Ubah Data Jenis Bestek	*Black box*
4	Data Bestek	Tambah Data Jenis Bestek	*Black box*
		Ubah Data Jenis Bestek	*Black box*
5	Stok Kayu	Tambah Data Stok Kayu	*Black box*
		Ubah Data Stok Kayu	*Black box*
6	Pengecekkan Stok	Cek Stok	*Black box*
		Pesan	*Black box*
7	Daftar Pesanan	Ubah Pesanan	*Black box*
8	Laporan Stok	Cek Laporan Stok	*Black box*
9	Laporan Pesanan	Cek Laporan Pesanan	*Black box*
10	Algoritma Greedy	Kombinasi kayu potong	*White Box*

IV.2.1 Pengujian *Black box*

Pengujian fungsionalitas yang digunakan adalah pengujian black box. Black box berfokus pada persyaratan fungsional dari sistem yang dibangun. Berikut adalah tabel pengujian black box pada aplikasi yang dijelaskan pada tabel IV.5 sampai dengan IV.13.

Tabel IV.5 Pengujian Login

Kelas Uji	Skenario Uji	Hasil yang Diharakan	Kesimpulan
Login	Memasukkan username dan password	Masuk kedalam aplikasi	[√] Berhasil [ ] Tidak Berhasil

Tabel IV.6 Pengujian Menu Utama

Kelas Uji	Skenario Uji	Hasil yang Diharapkan	Kesimpulan
Menu Utama	Memilih Jenis Bestek	Menampilkan form browseJenisBestek	[√] Berhasil [ ] Tidak Berhasil dak Berhasil
	Memilih Data Bestek	Menampilkan form browseBestek	
	Memilih Stok Kayu	Menampilkan form browseKayu	
	Memilih Pengecekkan Stok	Menampilkan form cekStok	
	Memilih Daftar Pesanan	Menampilkan form browsePesanan	
	Memilih Laporan Stok	Menampilkan form browseLaporanStok	
	Memilih Laporan Pesanan	Menampilkan form browseLaporanSPesanan	

Tabel IV.7 Pengujian Jenis Bestek

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan
Jenis Bestek	Memilih tombol tambah data jenis bestek	Menampilkan form tambah data jenis bestek	[√] Berhasil [ ] Tidak Berhasil
	Memilih tombol ubah data jenis bestek	Menampilkan form edit data jenis bestek	

Tabel IV.8 Pengujian Data Bestek

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan
Data Bestek	Memilih tombol tambah data bestek	Menampilkan form tambah data bestek	[√] Berhasil [ ] Tidak Berhasil
	Memilih tombol ubah data bestek	Menampilkan form edit data bestek	

Tabel IV.9 Pengujian Stok Kayu

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan
Stok Kayu	Memilih tombol tambah data stok kayu	Menampilkan form tambah data bestek	[√] Berhasil [ ] Tidak Berhasil
	Memilih tombol ubah data stok kayu	Menampilkan form edit data bestek	

Tabel IV.10 Pengujian Pengecekkan Stok

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan
Pengecekkan Stok	User memasukkan jumlah kayu bestek yang akan dipesan di field jumlah pada datagrid cekStok	Menampilkan pesan YA atau TIDAK di datagrid cekstok pada field terpenuhi, menampilkan hasil perhitungan pada datagrid pemenuhan pemesanan dan mengaktifkan tombol pesan	[√] Berhasil [ ] Tidak Berhasil
	User memasukkan data pemesan pada form pemesan	Menyimpan data pemesan pada tabel pemesanan	

Tabel IV.11 Pengujian Daftar Pemesan

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan
Daftar Pemesan	User menekan tombol daftar pesanan	Menampilkan data pesanan, data pemesanan detai dan data pemenuhan pemesanan	[√] Berhasil [ ] Tidak Berhasil
	User menekan tombol edit	Menampilkan form edit data pesanan	

Tabel IV.12 Pengujian Laporan Stok

Kelas Uji	Skenario Uji	Hasil yang Diharakan	Kesimpulan
Laporan Stok	Memasukkan tanggal awal dan tanggal akhir	Menampilkan laporan stok dari tanggal awal sampai tanggal akhir	[√] Berhasil [ ] Tidak Berhasil

Tabel IV.13 Pengujian Laporan Pesanan

Kelas Uji	Skenario Uji	Hasil yang Diharakan	Kesimpulan
Laporan Pesanan	Memasukkan tanggal awal dan tanggal akhir	Menampilkan laporan pesanan dari tanggal awal sampai tanggal akhir	[√] Berhasil [ ] Tidak Berhasil

IV.2.2 Pengujian White Box

Pengujian white box merupakan metode perancangan test case yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan test case. Adapun metode yang digunakan dalam pengujian white box adalah metode basis path. Metode basis path mengijinkan pendesain kasus uji untuk membuat perkiraan lojik yang kompleks dari desain prosedural dan menggunakan perkiraan ini untuk mendefinisikan aliran eksekusi.

a. Berikut adalah Berikut adalah pseodocode greedy yang dijelaskan pada tabel IV.14.

Tabel IV.14Pseoducode Greedy pada Aplikasi

	this.node = node;
	this.stok_id = stok_id;
	this.hasil = hasil;
	this.sisa = sisa;
	Int32 bestekPotong = 0;
	int node = 0;
	List<Greedy> hasilGreedy = new List<Greedy>();
	bool stokHabis = false;

1	while (bestekPotong < bestekPesanan.jumlah && !stokHabis) {
2	if (kayuBalokforBestek.Count > 0) {

3	Greedy greedy = new Greedy();
	node++;
	greedy.node = node;
	greedy.sisa = -1;

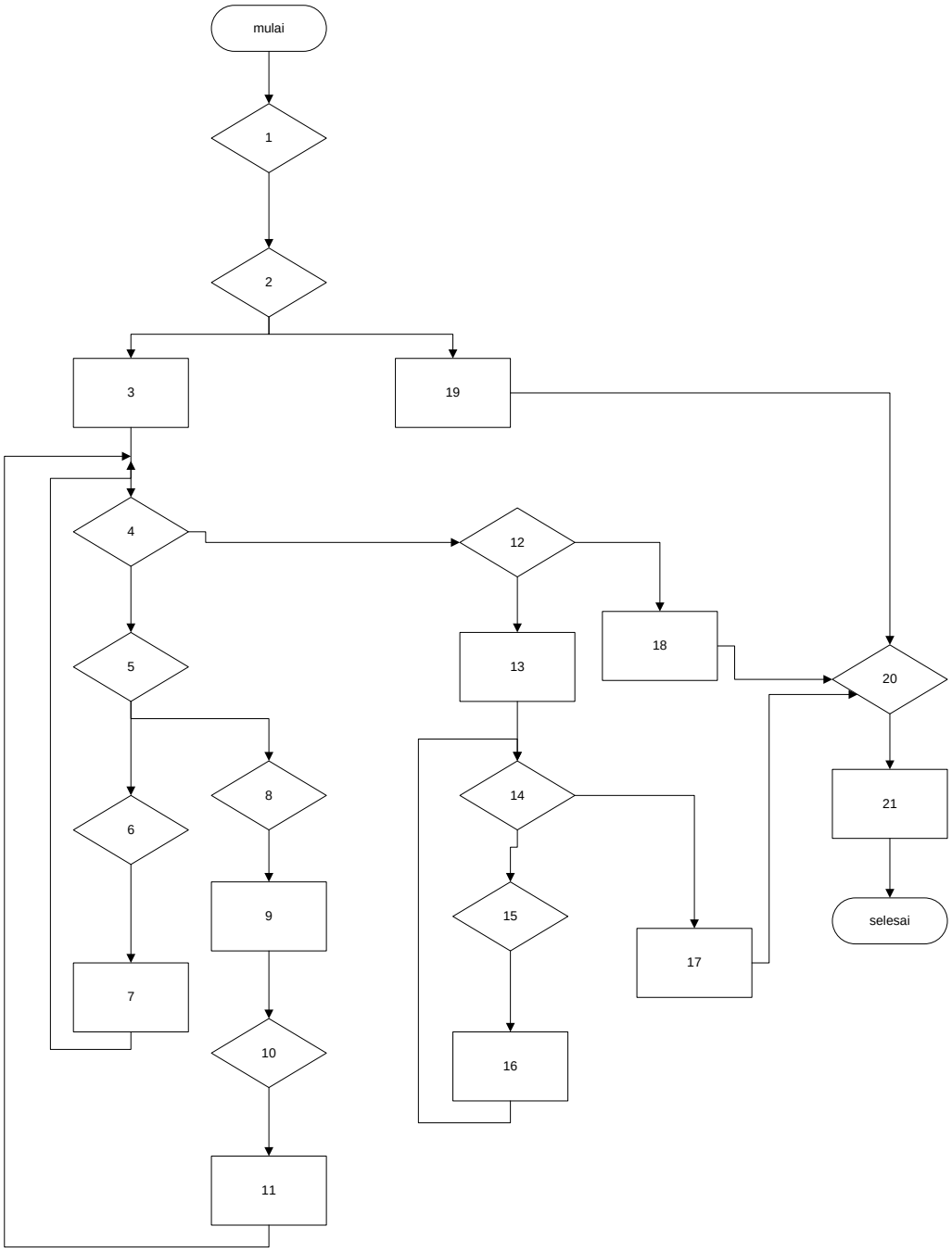
4	foreach (kayuBalok balok in kayuBalokforBestek) {
5	if ((balok.jumlah - balok.jumlahPakai) > 0) {
6	if (balok.hasil <= (bestekPesanan.jumlah - bestekPotong) && (greedy.sisa == -1 || balok.sisa < greedy.sisa)) {

7	greedy.sisa = balok.sisa;
	greedy.stok_id = balok.stok_id;
	greedy.hasil = balok.hasil;
	}

8	else if (balok.hasil > (bestekPesanan.jumlah - bestekPotong)) {
9	int volumeBestek = bestekPesanan.lebar * bestekPesanan.panjang * bestekPesanan.tinggi; int hasilTerpakai = bestekPesanan.jumlah - bestekPotong;
10	int sisaPakai = ((balok.hasil - hasilTerpakai) * volumeBestek) + balok.sisa; if ((greedy.sisa == -1 || balok.sisa < greedy.sisa))
11	{

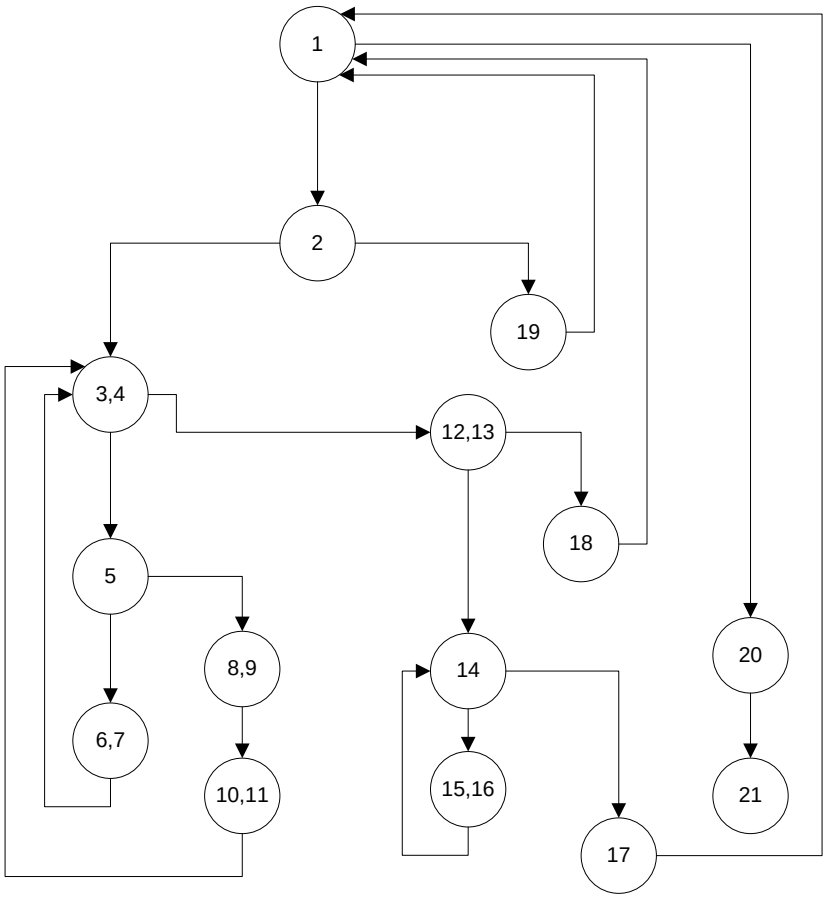
```
                        greedy.sisa = sisaPakai;
                        greedy.stok_id = balok.stok_id;
                        greedy.hasil = hasilTerpakai;
                       }
                      }
                     }
                    }
12                  if (greedy.sisa != -1)
                    {
13                    hasilGreedy.Add(greedy);
14                    foreach (kayuBalok balok in kayuBalokforBestek)
                      {
15                     if (balok.stok_id == greedy.stok_id)
                       {
16                      balok.jumlahPakai++;
                        break;
                       }
                      }
17                    bestekPotong += greedy.hasil;
                      }
18                    else
                      {
                       stokHabis = true;
                       break;
                     }
                    }
19                    else stokHabis = true;
                      }
20                     if(stokHabis)
                       {
21                      hasilGreedy.Clear();
                       }
            end;
```

Berdasarkan pseodocode greedy pada tabel IV.14, maka didapat flow chart yang dijelaskan pada gambar IV.15.



Gambar IV.15 Flow Chart Greedy

Berdasarkan flow chart greedy pada gambar IV.15, maka didapat flow graph yang dijelaskan pada gambar IV.16.



Gambar IV.16 Flow Graph Greedy

Berdasarkan flow graph pada gambar IV.16, maka dapat dihitung Cyclomatic complexity, yakni:

V(G) = E – N + 2

= 19 – 15 + 2

= 6

Dari perhitungan Cyclomatic complexity, didapatkan 6 independent path, yaitu:

Path 1 : 1,2,3,4,5,6,7,3,4,12,13,18,1,20,21

Path 2 : 1,20,21

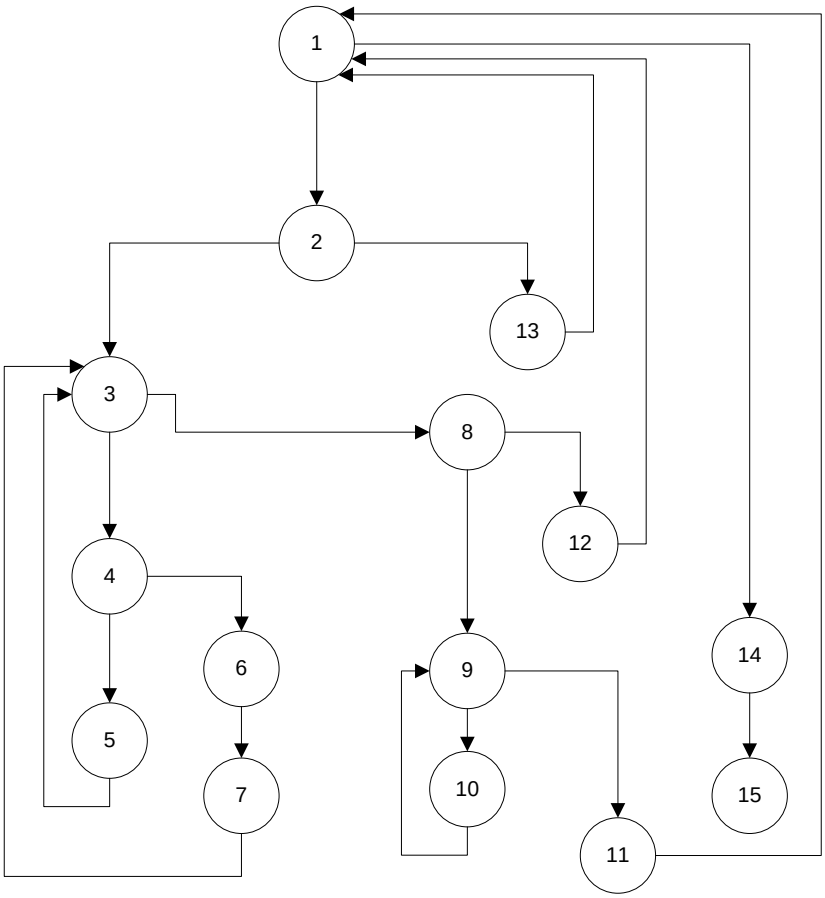
Path 3 : 11,2,19,1,20,21

Path 4 : 1,2,3,4,5,8,9,10,11,3,4,12,13,14,15,16,14,17,1,20,21

Path 5 : 1,2,3,4,12,13,18,1,20,21

Path 6 : 1,2,3,4,12,13,14,17,1,20,21

Uji coba basis path digunakan sebuah graph matrix. Graph matrix merupakan empat persegi yang mempunyai ukuran yang sama dengan jumlah node pada flowgraph. Flowgraph yang ada pada gambar IV.16akan dikonversi sesuai seperti gambar IV.17.



Gambar IV.17 Konversi Flowgraph

Graph matrix adalah matriks kotak yang besarnya sama dengan jumlah dari node di flow graph. Setiap baris dan kolom sesuai dengan node dan angka pada matriks mewakili dari koneksi setiap node dimana jika nilainya 1, maka antara node tersebut ada koneksi, sedangkan jika nilainya 0, maka antara node tersebut tidak memiliki koneksi. Adapun graph matrix pada flowgraph gambar IV.2 diterangkan pada tabel IV.15.

Tabel IV.15 Graph Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	predicate node
1		1												1		1
2			1										1			1
3				1				1								1
4					1	1										1
5			1													
6							1									
7			1													
8									1			1				1
9										1	1		1			1
10									1							
11	1															
12	1															
13	1															
14															1	
15																
Jumlah																6

Dari Graph matrix pada tabel IV.15, dapat diketahui bahwa predicate node adalah node 1 dan 2 dan jumlah predicate node dapat dihitung dengan rumus

V(G) = P + 1

= 6 + 1

= 7

IV.3 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung dilapangan, dengan cara mengujikan aplikasi langsung kepada user yang kemudian dilakukan wawancara kepada user mengenai kelayakan aplikasi yang dibuat.

IV.3.1 Skenario Pengujian Beta

Pengujian beta ini dilakukan dengan cara melakukan pengujian langsung kepada pihak-pihak yang terkait dengan penelitian ini. Pihak yang dimaksud adalah pihak-pihak yang menjadi narasumber awal yaitu pemimpin perusahaan.

Adapun beberapa poin yang diteliti adalah :

a. Tampilan antarmuka yang mudah digunakan.
b. Memudahkan penerimaan pemesanan kayu bestek.
c. Mengoptimalkan rendemen pada kayu.

d. Mempermudah pemimpin perusahaan menghitung pemotongan kayu bestek.
e. Kemudahan melakukkan pengecekkan stok kayu.
f. Kemudahan mendapatkan informasi pemotongan kayu.

IV.3.1.1 Skenario Pengujian Beta Pemimpin

Pengujian beta pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu ini menggunakan teknik wawancara kepada bapak Sigit Susanto selaku pemimpin perusahaan. Berikut ini adalah daftar pertanyaan hasil wawancara yang diberikan kepada pemimpin perusahaan.

1. Menurut Anda, apakah pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini mempermudah penerimaan pemesanan kayu bestek?
   Jawaban: pemimpin perusahaan mengatakan bahwa pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini sudah cukup mempermudah penerimaan pemesanan kayu bestek.
2. Menurut Anda, apakah pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini mempermudah pemimpin menghitung pemotongan kayu bestek?
   Jawaban : pemimpin perusahaan mengatakan bahwa pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini sudah cukup mempermudah pemimpin menghitung pemotongan kayu bestek, hanya saja ada beberapa hal yang harus diperjelas.
3. Menurut Anda, apakah pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini menghasilkan hasil sisa yang optimal?
   Jawaban: pemimpin perusahaan mengatakan bahwa pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini sudah cukup menghasilkan hasil sisa yang kurang optimal

4. Menurut Anda, apakah pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini memberikan kemudahan dalam pengecekkan stok kayu?

   Jawaban: pemimpin perusahaan mengatakan bahwa pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini sangat dipermudah dalam pengecekkan stok
5. Menurut Anda, apakah pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini memiliki tampilan antarmuka yang mudah digunakan?

   Jawaban: pemimpin perusahaan mengatakan bahwa pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini memiliki tampilan yang masih perlu diperbaiki.
6. Menurut Anda, apakah pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini mempermudah pemimpin untuk mendapatkan informasi pemotongan kayu?

   Jawaban: pemimpin perusahaan mengatakan bahwa pembangunan sistem informasi untuk mengoptimalkan rendemen pada kayu di P.D. Sukamaju ini sudah cukup membantu pemimpin perusahaan mendapatkan informasi pemotongan kayu.

IV.3.1.2 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta yang telah dilakukan, dapat diambil kesimpulan bahwa:

a. Sistem yang dibangun mempermudah penerimaan pemesanan bestek kayu
b. Sistem yang dibangun mempermudah pemimpin menghitung pemotongan kayu bestek, tapi ada beberapa bagian yang harus diperbaiki.
c. Sistem yang dibangun menghasilkan perhitungan yang kurang optimal.
d. Sistem yang dibangun sangat mempermudah pemimpin melakukan pengecekkan stok kayu.

e. Tampilan antarmuka kurang mudah dipergunakan, sehingga perlu perbaikan.
f. Sistem yang dibangun mempermudah pemimpin perusahaan untuk mendapatkan informasi pemotongan kayu.