

BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisa Sistem Yang Sedang Berjalan

Lamport's bakery algorithm merupakan solusi yang diajukan oleh Lamport untuk menyelesaikan masalah *mutual exclusion* untuk n proses. *Lamport's bakery algorithm* ini menerapkan algoritma proses antrian FIFO (*First In First Out*) yang berarti pembeli yang pertama datang akan memiliki nomor yang lebih kecil. Pembeli yang bernomor lebih kecil akan dilayani oleh pelayan terlebih dahulu.

Sedangkan, model antrian yang diterapkan oleh *Lamport's bakery algorithm* ini adalah model antrian MQMS (*Multi Queue Multi Server*), yang berarti bahwa terdapat n buah antrian yang akan dilayani oleh n orang pelayan, dimana n bernilai lebih dari satu.

III.1.1. Asumsi Dari Perangkat Lunak

Asumsi yang digunakan oleh perangkat lunak simulasi *Lamport's bakery algorithm* ini adalah sebagai berikut :

1. Hanya satu orang pembeli yang masuk dalam waktu tertentu. Dalam hal ini berarti setiap pembeli memiliki perbedaan waktu.
2. Jumlah pembelian dari setiap pembeli dicatat, namun jenis item tidak diperhatikan.

3. Satuan terkecil waktu yang diambil adalah satuan menit. Hal ini berarti bahwa waktu dalam perangkat lunak akan bergerak 1 menit setiap waktu komputer bergerak n milidetik.
4. Jumlah kasir dibatasi maksimal sebanyak 5 buah dan minimal sebanyak 2 buah.
5. Jumlah kursi tunggu dibatasi sebanyak 10 buah.
6. Jenis kue dibatasi sebanyak 5 jenis saja.
7. Jumlah item kue yang dapat disimpan maksimal 5 digit bilangan bulat positif. Jumlah item ini mewakili jumlah *resource* yang tersedia dalam sebuah aplikasi.
8. Jumlah item memiliki batas kritis yang dapat di-*input* secara manual, dengan batasan minimal 5 buah dan maksimal 20 buah. Batas kritis ini mewakili sisa *resource* yang tersedia dalam aplikasi. Apabila aplikasi telah mencapai titik kritis, maka aplikasi akan berjalan lebih lambat, sementara aplikasi menyediakan *resource* yang diperlukan.

Hal-hal yang akan dicatat dalam simulasi ini adalah sebagai berikut :

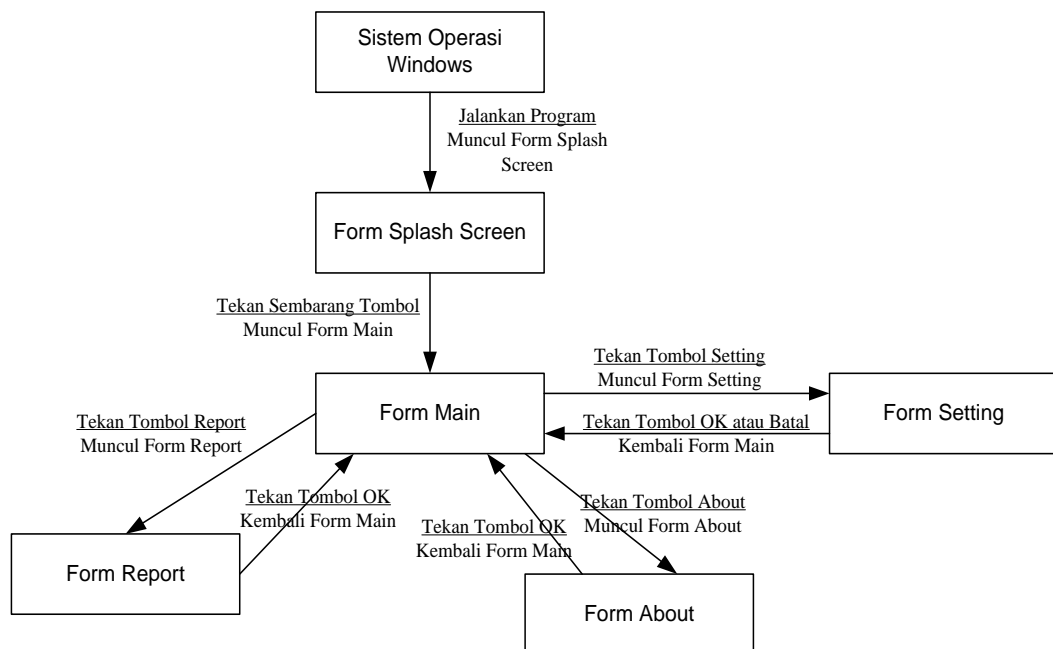
1. Lama pembeli menunggu di tempat tunggu (antrian).
2. Lama pembeli memilih kue.
3. Lama pembeli melakukan pembayaran.

III.1.2. Diagram Transisi State Dari Perangkat Lunak

Perangkat lunak simulasi *Lamport's Bakery Algorithm* ini dimulai dari pembukaan *form* 'Splash Screen' yang berisi data identitas dari penyusun. Setelah

itu, akan ditampilkan *form 'Main'* yang merupakan *form* inti dari perangkat lunak. *Form 'Main'* akan menampilkan simulasi dari *Lamport's Bakery Algorithm* dengan data-data keadaan awal dapat di-input pada *form 'Setting'* yang dapat dipanggil dengan meng-klik tombol '*Setting*'. Sedangkan, apabila ingin melihat laporan dari simulasi yang sedang dijalankan, maka dapat meng-klik tombol '*Report*' untuk memunculkan *form 'Report'*. Sementara, apabila meng-klik tombol '*About*' maka akan tampil *form 'About'* yang berisi keterangan mengenai perangkat lunak simulasi ini.

Proses kerja dari perangkat lunak ini dapat dilihat pada gambar berikut ini:

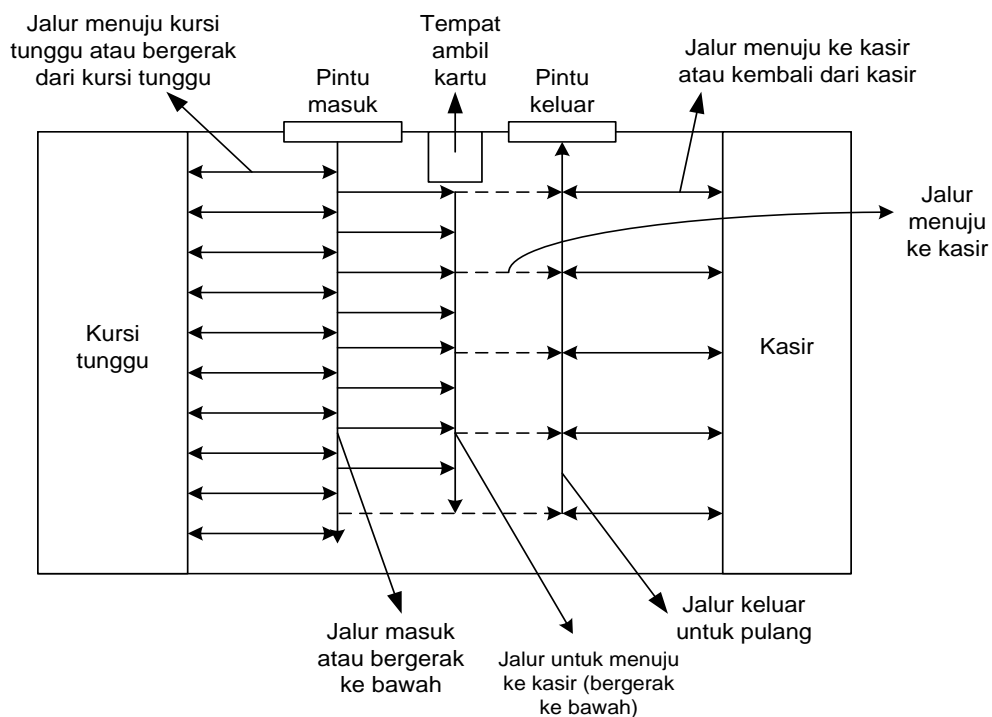


Gambar III.1 Diagram Transisi *State* Dari Perangkat Lunak

III.1.3. Proses Animasi Pada Perangkat Lunak

Perangkat lunak simulasi *Lamport's Bakery Algorithm* ini menggunakan teknik animasi pergantian gambar dan pergantian posisi. Kedua teknik ini

digunakan terutama pada saat pembeli berjalan. Saat pembeli melangkah, digunakan 2 buah gambar secara bergantian, yaitu pembeli melangkah dengan kaki kiri dan pembeli melangkah dengan kaki kanan. Caranya adalah sebagai berikut, pertama-tama, tampilkan gambar pembeli melangkah dengan kaki kiri, berhenti selama waktu tertentu, kemudian ganti dengan gambar pembeli melangkah dengan kaki kanan.



Gambar III.2 Jalur Pergerakan Pembeli Dalam Sistem

Arah pergerakan pembeli dalam sistem dapat dilihat pada gambar III.2 di atas. Teknik animasi yang digunakan adalah teknik pergantian gambar. Gambar-gambar yang digunakan akan dirincikan dalam sub bab perancangan gambar. Ketika pembeli bergerak ke bawah, maka digunakan pembeli melangkah menuju ke bagian bawah layar komputer yaitu kelihatan bagian depan dari badan pembeli.

Sementara, apabila pembeli bergerak ke atas, maka digunakan pembeli melangkah menuju ke bagian atas layar computer, yaitu hanya kelihatan bagian belakang dari badan pembeli. Apabila pembeli bergerak ke kiri atau kanan, maka diambil gambar pembeli menuju ke kiri atau kanan dari layar computer, yaitu hanya kelihatan bagian samping kiri atau kanan dari badan pembeli.










III.2. Evaluasi Sistem Yang Berjalan

Perangkat lunak simulasi ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Studio 2010* dengan didukung oleh aplikasi *Adobe Photoshop C.S* untuk merancang dan menyimpan gambar-gambar yang diperlukan oleh perangkat lunak.

III.2.1. Perancangan Gambar

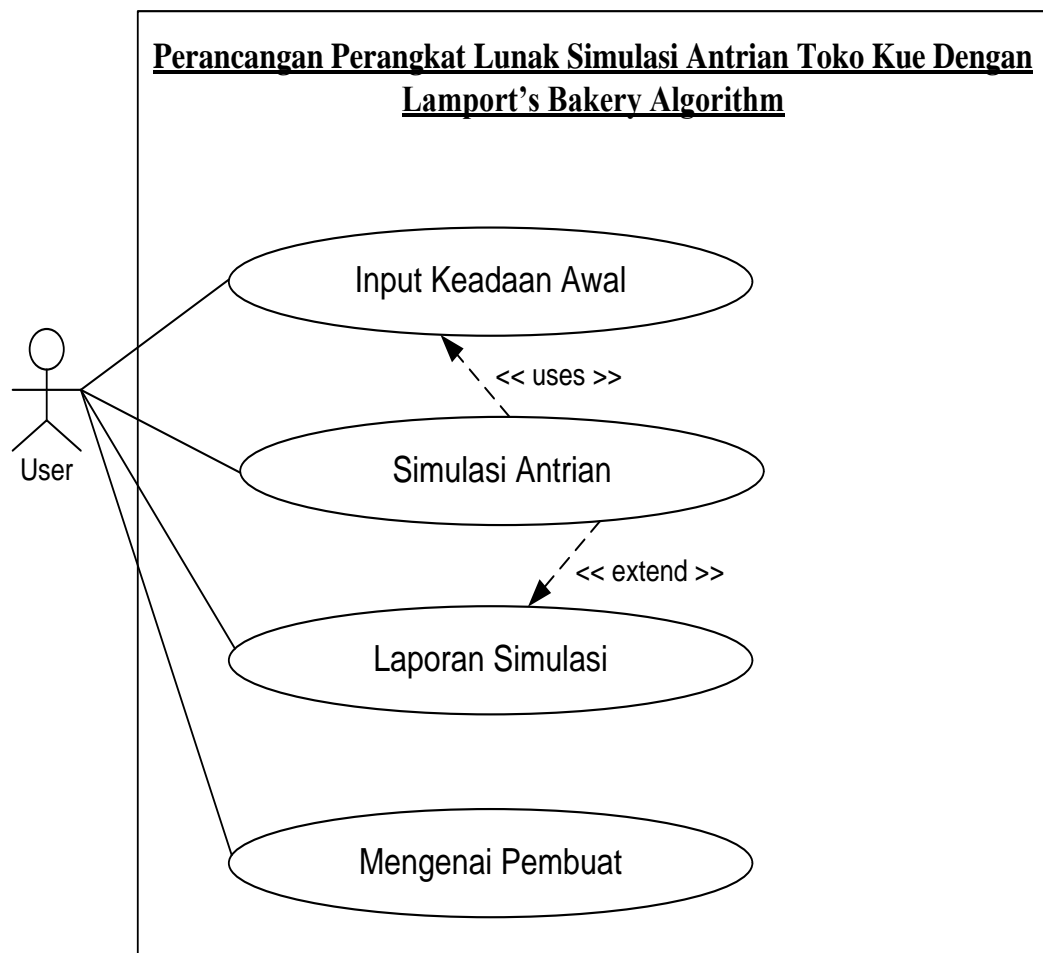
Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan proses simulasi antrian toko kue. Dengan *storyboard* ini, penulis dapat menyampaikan jalannya proses simulasi dengan menggunakan tabel gambar berikut ini :

Tabel III.1. *Storyboard* Simulasi Antrian

No.	<i>Storyboard</i> Toko Kue	Keterangan
1		Jenis burger biasa yang akan dijual pada toko kue.
2		Jenis burger berbentuk pita yang tersedia di toko kue.
3		Jenis <i>cake</i> yang ada di toko kue.
4		Jenis roti yang ada pada antrian toko kue.
5		Jenis kue yang ada pada antrian toko kue.
6		Kasir toko kue yang akan melayani antrian <i>customer</i> .
7		<i>Customer</i> A menghadap ke samping kanan.
8		<i>Customer</i> B menghadap ke samping kiri dengan membawa bungkusan.
9		<i>Customer</i> C menghadap ke samping kiri dengan membawa bungkusan.

III.2.2. Use Case Diagram

Use case diagram digunakan untuk menggambarkan visualisasi animasi yang sedang berlangsung pada antrian toko kue. Perancangan *use case* dapat dilihat pada gambar berikut :



Gambar III.3. Rancangan *Use Case* Diagram

III.2.3. Algoritma Pengaturan Waktu

Algoritma ini berfungsi untuk mengatur waktu yang berjalan. Prosedur kerja dari algoritma ini dapat dijabarkan sebagai berikut :

1. Algoritma Penjumlahan Waktu Dengan Angka

Fungsi FTmbh(pcDate, pnInc)

nJam \leftarrow Ambil karakter kiri dari (pcDate) sebanyak 2 buah

nMenit \leftarrow Ambil karakter kanan dari (pcDate) sebanyak 2 buah

nMenit \leftarrow nMenit + pnInc

Jika nMenit \geq 60 maka

nJam \leftarrow nJam + 1

nMenit \leftarrow nMenit - 60

Jika tidak, jika nJam \geq 24 maka

nJam \leftarrow nJam - 24

FTmbh \leftarrow Format(nJam, "00") + " : " + Format(nMenit, "00")

End Fungsi

2. Algoritma Penjumlahan Waktu Dengan Waktu

Fungsi FAdd(pcDateBefore, pcDateAfter)

nJam \leftarrow Ambil karakter kiri dari (pcDateBefore) sebanyak 2 buah

nMenit \leftarrow Ambil karakter kanan dari (pcDateBefore) sebanyak 2 buah

nJam1 \leftarrow Ambil karakter kiri dari (pcDateAfter) sebanyak 2 buah

nMenit1 \leftarrow Ambil karakter kanan dari (pcDateAfter) sebanyak 2 buah

$$nTMenit \leftarrow nMenit + nMenit1$$

Jika $nTMenit > 60$ maka

$$nInc \leftarrow nTMenit \setminus 60$$

$$nTMenit \leftarrow nTMenit \text{ Mod } 60$$

$$nTJam \leftarrow nJam + nJam1 + nInc$$

$$FAdd \leftarrow \text{Format}(nTJam, "00") + " : " + \text{Format}(nTMenit, "00")$$

End Fungsi

3. Algoritma Pengurangan Waktu Dengan Waktu

Fungsi FSlsh(pcDateBefore, pcDateAfter)

$$nJam \leftarrow \text{Ambil karakter kiri}(pcDateBefore) \text{ sebanyak } 2 \text{ buah}$$

$$nMenit \leftarrow \text{Ambil karakter kanan}(pcDateBefore) \text{ sebanyak } 2 \text{ buah}$$

$$nJam1 \leftarrow \text{Ambil karakter kiri}(pcDateAfter) \text{ sebanyak } 2 \text{ buah}$$

$$nMenit1 \leftarrow \text{Ambil karakter kanan}(pcDateAfter) \text{ sebanyak } 2 \text{ buah}$$

Jika $nMenit1 < nMenit$ maka

$$nJam1 \leftarrow nJam1 - 1$$

$$nMenit1 \leftarrow nMenit1 + 60$$

Jika tidak, jika $nJam1 < nJam$ maka

$$nJam1 \leftarrow nJam1 + 24$$

$$FSlsh \leftarrow \text{Format}(nJam1 - nJam, "00") + " : " + \text{Format}(nMenit1 - nMenit, "00")$$

End Fungsi

III.2.4. Algoritma Pengaturan Waktu Kedatangan Pembeli

Waktu kedatangan pembeli diatur secara *random* (acak) oleh komputer. Prosedur kerja untuk mengatur waktu kedatangan pembeli dapat dijabarkan sebagai berikut :

'Ambil Nilai Random

$$nCreateCust \leftarrow \text{Round}(\text{Rnd} * 100)$$

'Create customer jika ada kursi kosong

Jika $(nCreateCust \text{ Mod } (14 - nKursi) \leftarrow 0)$ dan $bKursi$ maka

$$nCust \leftarrow nCust + 1$$

Jika $\text{imgOrang.UBound} < nCust$ maka

$$\text{Load imgOrang}(nCust)$$

'Ambil Tipe Orang

$$nTipe \leftarrow \text{Round}(\text{Rnd} * 3) + 1$$

$$\text{Cust}(nCust).W_Datang \leftarrow \text{txtWaktu}.Text$$

$$\text{Cust}(nCust).Selesai \leftarrow \text{False}$$

$$\text{Cust}(nCust).Proses \leftarrow \text{"Jalan"}$$

'Ambil Lama Service

Randomize Timer

Jika nTipe ≤ 2 maka

$$\text{Cust}(n\text{Cust}).\text{Lama_Srvs} \leftarrow \text{Round}(\text{Rnd} * 10) + 5$$

Jika tidak, maka

$$\text{Cust}(n\text{Cust}).\text{Lama_Srvs} \leftarrow \text{Round}(\text{Rnd} * 15) + 10$$

$$\text{Cust}(n\text{Cust}).\text{W_Antar_Kedgn} \leftarrow \text{Iif}(n\text{Cust} = 1, "00 : 00",$$

$$\text{FSIsh}(\text{Cust}(n\text{Cust}).\text{W_Datang}, \text{Cust}(n\text{Cust} - 1).\text{W_Datang}))$$

III.2.5. Algoritma Pengecekan Giliran

Algoritma ini berfungsi untuk melakukan pengecekan apakah pembeli tersebut telah mendapatkan giliran untuk dilayani atau belum. Prosedur kerja dari algoritma ini dapat dijabarkan sebagai berikut :

Fungsi IsGiliran(pnIndexCust)

Untuk nF \leftarrow 1 sampai Indeks tertinggi(imgOrang)

Jika Cust(nF).Proses \leftarrow "JALAN KELUAR" maka

$$n\text{Index} \leftarrow n\text{F}$$

$$n\text{Index} \leftarrow n\text{Index} + 1$$

Jika pnIndexCust \leftarrow nIndex Then IsGiliran \leftarrow True

End Fungsi

III.2.6. Algoritma Pengecekan Kursi Kosong

Algoritma ini berfungsi untuk mengecek apakah masih ada kursi kosong atau tidak. Jika ada, maka akan dicari kursi kosong, mulai dari nomor indeks kursi terkecil. Prosedur kerja dari algoritma ini dapat dijabarkan sebagai berikut :

Fungsi FKursiKosong()

Untuk $nA \leftarrow 1$ sampai $nKursi$

Jika $Kursi(nA) \leftarrow 0$ maka

$FKursiKosong \leftarrow nA$

Keluar dari fungsi

End Fungsi

III.2.7. Algoritma Pengecekan Depan Pintu

Algoritma ini berfungsi untuk mengecek apakah ada *customer* yang melewati pintu atau tidak. Prosedur kerja dari algoritma ini dapat dijabarkan sebagai berikut :

Fungsi FCekDepanPintu()

Untuk $nC \leftarrow 1$ sampai Indeks array tertinggi(*imgOrang*)

Jika $Cust(nC).PosX \geq 11350$ dan $Cust(nC).PosX \leq 11450$ maka

$FCekDepanPintu \leftarrow True$

Keluar dari fungsi

End Fungsi

III.2.8. Algoritma Pengaturan Gambar Pembeli Agar Tidak Bertindih

Algoritma ini berfungsi untuk mengatur agar objek pembeli tidak bertabrakan. Prosedur kerja dari algoritma ini dapat dijabarkan sebagai berikut :

Fungsi IsCrash(pnIndex, pnTX, pnTY, pcTX, pcTY)

Untuk I = 1 sampai indeks array tertinggi dari picOrang

Jika picOrang(I).Visible dan $I \neq pnIndex$ maka

Jika $Abs(picOrang(pnIndex).Left - picOrang(I).Left) <$

$picOrang(I).Width + pnTX$ dan $Abs(picOrang(pnIndex).Top -$

$picOrang(I).Top) < picOrang(I).Height + pnTY$ maka

Jika pcTX = "" dan pcTY = "" maka

IsCrash = True

Jika tidak, jika pcTX = "-" dan

$picOrang(pnIndex).Left - picOrang(I).Left < 0$ maka

IsCrash = True

Jika tidak, jika pcTX = "+" dan

$picOrang(pnIndex).Left - picOrang(I).Left > 0$ maka

IsCrash = True

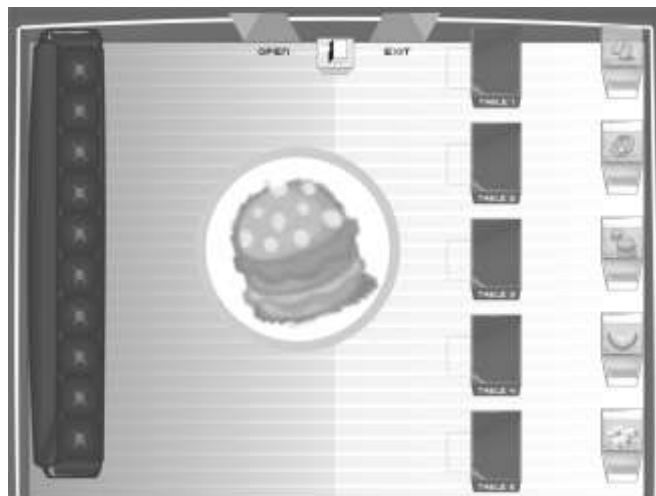
Jika tidak, jika pcTY = "-" dan

$picOrang(pnIndex).Top - picOrang(I).Top < 0$ maka

IsCrash = True

Jika tidak, jika $pcTY = "+"$ dan
 $picOrang(pnIndex).Top - picOrang(I).Top > 0$ maka
 $IsCrash = True$
Keluar dari fungsi
 $IsCrash = False$
End Fungsi

III.2.9. Penggunaan Komponen



Gambar III.4. Toko Kue

Gambar *background* di atas berupa gambar toko kue, yang dirancang dengan menggunakan Adobe Photoshop C.S, kemudian disimpan ke dalam bentuk *file *.gif*.

Perangkat lunak simulasi ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Studio 2010* dengan beberapa komponen standar, seperti :

1. *Command Button* yang digunakan sebagai tombol eksekusi.
2. *Picture Box* dan *Image* yang digunakan untuk menampilkan gambar.
3. *Label* yang digunakan untuk menampilkan keterangan tambahan.
4. *Shape* yang berfungsi sebagai dekorasi tambahan pada *form*.
5. *Timer* yang digunakan untuk mengatur animasi gambar.

III.3. Desain Sistem

III.3.1. *Form Splash Screen*

Form ini merupakan *form* pembuka (awal) dari perangkat lunak. *Form* ini akan ditutup apabila dilakukan penekanan sembarang tombol pada *keyboard*.

Rancangan tampilan dari *form* 'Main' ini dapat dilihat pada gambar berikut ini :



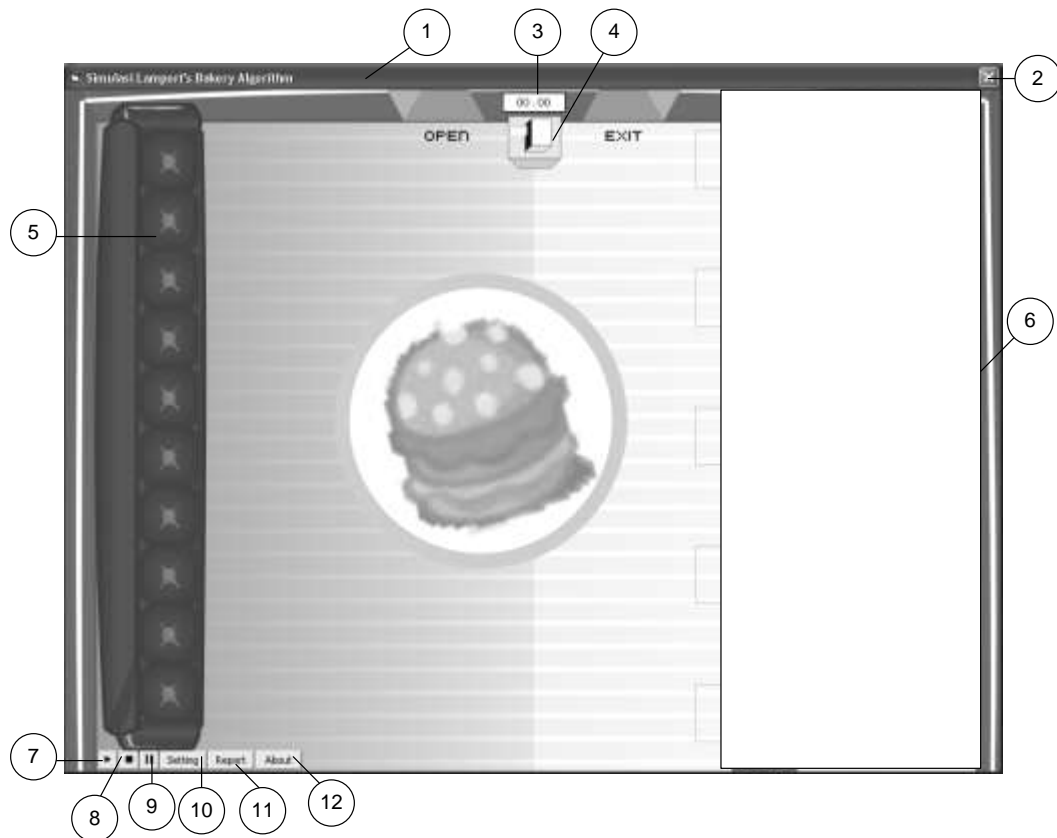
Gambar III.5. Rancangan *Form* 'Splash Screen'

Keterangan :

- 1 : gambar logo perangkat lunak
- 2 : nama perangkat lunak

III.3.2. Form Main

Form ini merupakan *form* inti dari perangkat lunak yang berfungsi untuk menghubungkan semua *form* yang ada pada perangkat lunak. Rancangan tampilan dari *form* 'Main' ini dapat dilihat pada gambar berikut ini :



Gambar III.6. Rancangan Form 'Main'

Keterangan :

- 1 : *title bar*, bertuliskan 'Simulasi Langer's Bakery Algorithm'.
- 2 : tombol 'x', untuk menutup *form*.
- 3 : daerah tampilan waktu simulasi.
- 4 : tempat pengambilan kartu.

- 5 : tempat tunggu pembeli.
- 6 : daerah tampilan pelayan.
- 7 : tombol '*Play*', berfungsi untuk memulai / melanjutkan proses simulasi.
- 8 : tombol '*Stop*', berfungsi untuk menghentikan proses simulasi secara permanen.
- 9 : tombol '*Pause*', berfungsi untuk menghentikan proses simulasi untuk sementara.
- 10 : tombol '*Setting*', berfungsi untuk menampilkan *form* '*Setting*'.
- 11 : tombol '*Report*', berfungsi untuk menampilkan *form* '*Report*'.
- 12 : tombol '*About*', berfungsi untuk menampilkan *form* '*About*'.

III.3.3. Form Setting

Form ini berfungsi untuk melakukan proses pengaturan *setting* keadaan awal yang diperlukan dalam proses simulasi. Rancangan tampilan dari *form* '*Setting*' ini dapat dilihat pada gambar berikut ini :

The image shows a dialog box titled 'Setting Keadaan Awal ...'. It contains several input fields and buttons. The fields are: 'Mulai Simulasi' (12:51), 'Batas kritis item 1' (5 buah), 'Batas kritis item 2' (5 buah), 'Batas kritis item 3' (5 buah), 'Batas kritis item 4' (5 buah), 'Jumlah pelayan' (5 buah), 'Selesai Simulasi' (18:00), and 'Kecepatan Animasi' (1000 milisekon). At the bottom right, there are 'OK' and 'Batal' buttons. Numbered callouts (1-12) point to the title bar, close button, 'Mulai Simulasi' field, 'Batas kritis item 1' field, 'Batas kritis item 2' field, 'Batas kritis item 3' field, 'Batas kritis item 4' field, 'Jumlah pelayan' field, 'Selesai Simulasi' field, 'Kecepatan Animasi' field, 'OK' button, and 'Batal' button respectively.

Gambar III.7. Rancangan Form 'Setting'

Keterangan :

- 1 : *title bar*, bertuliskan 'Setting Keadaan Awal'.
- 2 : tombol 'x', untuk menutup *form*.
- 3 : *textbox* 'Mulai Simulasi' berfungsi sebagai tempat penginputan waktu mulai simulasi.
- 4 : *textbox* 'Batas kritis item 1' berfungsi sebagai tempat penginputan batas kritis item 1.
- 5 : *textbox* 'Batas kritis item 2' berfungsi sebagai tempat penginputan batas kritis item 2.

- 6 : *textbox* 'Batas kritis item 3' berfungsi sebagai tempat penginputan batas kritis item 3.
- 7 : *textbox* 'Batas kritis item 4' berfungsi sebagai tempat penginputan batas kritis item 4.
- 8 : *textbox* 'Jumlah pelayan' berfungsi sebagai tempat penginputan jumlah pelayan.
- 9 : *textbox* 'Selesai Simulasi' berfungsi sebagai tempat penginputan waktu selesai simulasi.
- 10 : *textbox* 'Kecepatan animasi' berfungsi sebagai tempat penginputan kecepatan simulasi.
- 11 : tombol 'OK' berfungsi untuk menyimpan data yang di-*input* dan kembali ke *form* 'Main'.
- 12 : tombol 'Batal' berfungsi untuk membatalkan data yang di-*input* dan kembali ke *form* 'Main'.

III.3.4. Form Report

Form ini berfungsi untuk menampilkan laporan dari proses simulasi.

Rancangan tampilan dari *form* 'Report' ini dapat dilihat pada gambar berikut ini :



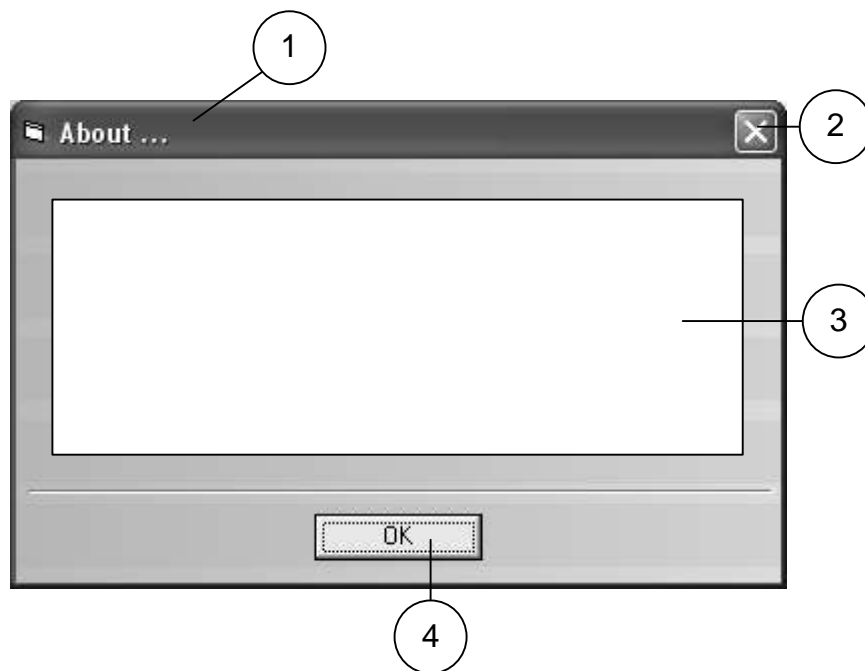
Gambar III.8. Rancangan *Form* ‘*Report*’

Keterangan :

- 1 : *title bar*, bertuliskan ‘*Report*’.
- 2 : tombol ‘x’, untuk menutup *form*.
- 3 : tabel ‘Tabel Data Simulasi’ berfungsi sebagai tempat untuk menampilkan data proses simulasi.
- 4 : tombol ‘Refresh’ berfungsi untuk membaca ulang data proses simulasi.
- 5 : tombol ‘Keluar’ berfungsi untuk menutup *form* dan kembali ke *form* ‘Main’.

III.3.5. *Form About*

Form ini berfungsi untuk menampilkan keterangan mengenai perangkat lunak simulasi ini. Rancangan tampilan dari *form* ‘*About*’ ini dapat dilihat pada gambar berikut ini :



Gambar III.9. Rancangan *Form* 'About'

Keterangan :

- 1 : *title bar*, bertuliskan 'About'.
- 2 : tombol 'x', untuk menutup *form*.
- 3 : daerah tampilan keterangan mengenai perangkat lunak.
- 4 : tombol 'OK' berfungsi untuk menutup *form* dan kembali ke *form* 'Main'.