

BAB VI

PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian dan analisis terhadap sistem yang telah dibangun. Pengujian yang dilakukan meliputi pengujian koneksi *database*, pengujian proses login, pengujian Aplikasi Admin, pengujian Aplikasi Operator Counter dan pengujian pembacaan barcode. Proses analisis dilakukan untuk mengetahui unjuk kerja dari sistem perangkat lunak yang sedang dibangun

6.1 Pengujian Koneksi Database

a. Tujuan

Pengujian ini dilakukan untuk memastikan bahwa komputer *client* dapat melakukan koneksi dengan *database* MySQL yang berada pada komputer *server*.

b. Spesifikasi komputer

- Prosesor AMD Athlon tm XP 1700+ 1.47GHz.
- Sistem Operasi Microsoft Windows XP Professional SP 2.
- Alamat IP Komputer *client* 192.10.10.1
- Alamat IP Komputer *server* 192.10.10.3

c. Software Aplikasi

- *Server database MySQL* 4.0.13-nt

d. Prosedur Pengujian

- Menjalankan Command Prompt dari [Start|Run|Open:cmd.exe|.
- Menampilkan koneksi yang sedang aktif pada komputer *client* sebelum melakukan koneksi pada *database* MySQL dengan memberikan perintah:

```
C:\mysql\bin>netstat -an
```

- Melakukan koneksi pada *server database* MySQL melalui komputer *client* dengan memberikan perintah:

```
C:\mysql\bin>mysql -u root -p -h 192.10.10.3
```

- Menampilkan koneksi yang sedang aktif pada komputer *client* setelah melakukan koneksi pada *database* MySQL dengan memberikan perintah:

```
C:\mysql\bin>netstat -an
```

e. Hasil Pengujian dan Analisa

Gambar 6.1 memperlihatkan koneksi yang aktif pada komputer *client* sebelum melakukan koneksi pada *database* MySQL.

```
C:\mysql\bin>netstat -an
Active Connections

Proto Local Address           Foreign Address         State
TCP   0.0.0.0:135              0.0.0.0:0               LISTENING
TCP   0.0.0.0:445              0.0.0.0:0               LISTENING
TCP   0.0.0.0:1030             0.0.0.0:0               LISTENING
TCP   0.0.0.0:5000             0.0.0.0:0               LISTENING
TCP   0.0.0.0:50300            0.0.0.0:0               LISTENING
TCP   192.10.10.1:139          0.0.0.0:0               LISTENING
TCP   192.10.10.1:1161        0.0.0.0:0               LISTENING
TCP   192.10.10.1:9967        0.0.0.0:0               LISTENING
UDP   0.0.0.0:135              *:*                      *:*
UDP   0.0.0.0:445              *:*                      *:*
UDP   0.0.0.0:500              *:*                      *:*
UDP   0.0.0.0:1025             *:*                      *:*
UDP   0.0.0.0:1026             *:*                      *:*
UDP   0.0.0.0:1300             *:*                      *:*
UDP   127.0.0.1:123            *:*                      *:*
UDP   127.0.0.1:1245           *:*                      *:*
UDP   127.0.0.1:1900           *:*                      *:*
UDP   192.10.10.1:123         *:*                      *:*
UDP   192.10.10.1:137         *:*                      *:*
UDP   192.10.10.1:138         *:*                      *:*
UDP   192.10.10.1:1900        *:*                      *:*
UDP   192.10.10.1:13229       *:*                      *:*
UDP   192.10.10.1:47333       *:*                      *:*
```

Gambar 6.1. Daftar Koneksi Pada Komputer Client Sebelum Melakukan Koneksi dengan Database MySQL

Sumber: Pengujian Dan Analisis

Gambar 6.1 menunjukkan hasil dari penggunaan perintah ‘netstat -an’, perintah tersebut digunakan untuk menampilkan semua koneksi yang terdapat pada komputer *client*. Parameter option ‘-an’ digunakan untuk menampilkan semua jenis koneksi dan *port* yang sedang melakukan *listening* dengan memperlihatkan alamat IP komputer dan *port* yang digunakan dalam bentuk numerik. Komputer *client* akan menjalankan perintah ‘mysql -u root -p -h 192.10.10.3’, perintah tersebut digunakan untuk melakukan koneksi dengan *database* MySQL yang terdapat pada komputer dengan alamat IP 192.10.10.3. *User* yang digunakan adalah *root* dengan password kosong. Koneksi antara

komputer *client* dengan *database* MySQL yang telah aktif diperlihatkan dalam Gambar 6.2.

```
C:\Documents and Settings\asisten>netstat -an
```

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1030	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1070	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5000	0.0.0.0:0	LISTENING
TCP	0.0.0.0:50300	0.0.0.0:0	LISTENING
TCP	192.10.10.1:1330	0.0.0.0:0	LISTENING
TCP	192.10.10.1:1070	192.10.10.3:3306	ESTABLISHED
TCP	192.10.10.1:13368	0.0.0.0:0	LISTENING
UDP	0.0.0.0:135	**:	**:
UDP	0.0.0.0:445	**:	**:
UDP	0.0.0.0:5000	**:	**:
UDP	0.0.0.0:1025	**:	**:
UDP	0.0.0.0:1026	**:	**:
UDP	0.0.0.0:1071	**:	**:
UDP	127.0.0.1:123	**:	**:
UDP	127.0.0.1:1900	**:	**:
UDP	192.10.10.1:123	**:	**:
UDP	192.10.10.1:137	**:	**:
UDP	192.10.10.1:138	**:	**:
UDP	192.10.10.1:1900	**:	**:
UDP	192.10.10.1:8708	**:	**:
UDP	192.10.10.1:59163	**:	**:

Gambar 6.2. Daftar Koneksi Pada Komputer Client Setelah Melakukan Koneksi dengan Database MySQL

Sumber: Pengujian Dan Analisis

Dalam Gambar 6.2 memperlihatkan koneksi yang terjadi antara komputer *client* dengan alamat 192.10.10.1 dengan *database* MySQL dengan alamat 192.10.10.3 pada port 3306.

6.2 Pengujian Proses Login

a. Tujuan

Pengujian dilakukan untuk mengetahui apakah aplikasi *login* dapat melakukan proses *login* yaitu dengan memberikan masukan pada *login* untuk mengetahui respon sistem saat diberikan *login* yang benar dan salah. Pengujian yang dilakukan akan dibandingkan dengan data pada *database* untuk mengetahui validitas dari sistem.

b. Prosedur Pengujian

1. Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court yaitu FoodCourtSys.exe.
2. Menu Login akan keluar seperti yang ditunjukkan dalam Gambar 6.3:



Gambar 6.3 Halaman Menu Login
Sumber : *Pengujian Dan Analisis*

Data login admin yang dimasukkan sebagai berikut:

- *Username="sysadmin"* dan *password="test"*, merupakan data login yang ada dalam *database* (bernilai benar).
- *Username="sysadmin"* dan *password="sembarang"*, merupakan data login tidak sesuai atau tidak ada dalam *database* (bernilai salah).

Apabila yang melakukan login adalah operator counter, maka data login operator counter yang dimasukkan sebagai berikut:

- *Username="counter1"* dan *password="counter1"*, merupakan data login yang ada dalam *database* (bernilai benar).
- *Username=" counter1"* dan *password="salah"*, merupakan data login tidak sesuai atau tidak ada dalam *database* (bernilai salah).

Login yang dilakukan oleh operator counter ditunjukkan dalam Gambar 6.4.



Gambar 6.4 Login Operator Counter
Sumber : *Pengujian Dan Analisis*

c. Hasil Pengujian dan Analisis

- Pada hasil login admin dengan *username* dan *password* yang bernilai benar (sesuai dengan data dalam *database*), mempunyai keluaran berupa

halaman menu utama admin. Aplikasi halaman menu utama admin ditunjukkan dalam Gambar 6.5.



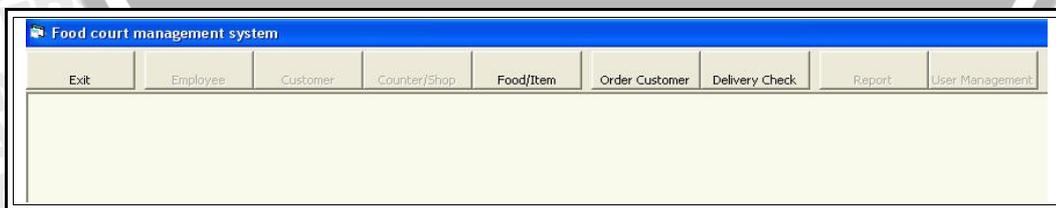
Gambar 6.5 Halaman Menu Utama Admin
 Sumber : *Pengujian Dan Analisis*

Pada hasil *login* admin yang bernilai salah (data tidak sesuai dengan data dalam *database*), akan tampil pesan "Access Denied". Klik OK untuk login kembali. Tampilan pesan kesalahan *login* salah ditunjukkan dalam Gambar 6.6.



Gambar 6.6 Halaman Pesan Kesalahan *Login* Salah
 Sumber : *Pengujian Dan Analisis*

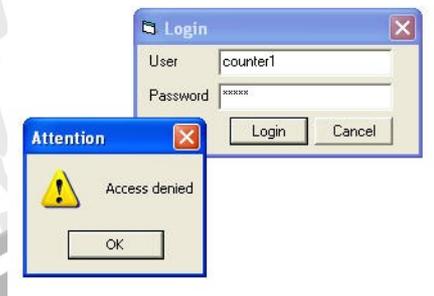
- Pada hasil *login* operator counter dengan *username* dan *password* yang bernilai benar (sesuai dengan data dalam *database*), mempunyai keluaran berupa halaman menu utama operator counter. Aplikasi halaman menu utama operator counter ditunjukkan dalam Gambar 6.7.



Gambar 6.7 Halaman Menu Utama Operator Counter
 Sumber : *Pengujian Dan Analisis*

Pada hasil *login* operator counter yang bernilai salah (data tidak sesuai dengan data dalam *database*), akan tampil pesan "Access Denied". Klik OK untuk

login kembali. Tampilan pesan kesalahan *login* salah ditunjukkan dalam Gambar 6.8.



Gambar 6.8 Tampilan Pesan Kesalahan *Login* Salah
Sumber : *Pengujian Dan Analisis*

6.3 Pengujian Aplikasi Admin

6.3.1 Pengujian Proses Registrasi Employee

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data employee dapat teregister dengan baik dan tersimpan pada `tb_employee` dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi**|`FoodCourtSys.exe` |.
- Memilih menu employee pada Sistem Transaksi Pembayaran Food Court.
- Memilih menu |**Employee Registration**| pada Employee Master Data
- Memasukan data employee melalui Sistem Transaksi Pembayaran Food Court
- Menampilkan data employee setelah data disimpan pada tabel `tb_employee` dengan memberikan perintah :

```
SELECT emp_id, emp_name, address, position, gender  
FROM tb_employee
```

c. Hasil Pengujian dan Analisis

Antarmuka Registrasi Employee pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.9:

EMP_ID	EMP_NAME	ADDRESS	POSITION	GENDER	E-MAIL	DEL?	CREATE BY	CREATE DATE
501234	Andrel	Jl. Merpati 53	Waitres	F	andrel@yahoo.com	N	sysadmin	17/09/2008 16:30:4
500500	Kurosaki Ichigo	Jl. Serenetai	Sales	M			sysadmin	18/06/2008 13:54:5

Gambar 6.9 Antarmuka Registrasi Employee
Sumber : *Pengujian Dan Analisis*

Data employee dimasukkan dengan memberikan parameter yang dibutuhkan pada Sistem Transaksi Pembayaran Food Court. Contoh data yang dimasukkan terdapat pada Tabel 6.1.

Tabel 6.1 Isi Pengujian Data Employee

Emp. ID	Emp. Name	Address	Position	Gender
501234	Andrel	Jl.Merpati 53	Waitres	Female
500500	Kurosaki Ichigo	Jl. Serenetai	Sales	Male
505566	Lulu	Jl.Ditempat123	Admin	Female

Hasil data yang dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran FoodCourt diperlihatkan dalam Gambar 6.10.



```
mysql> select emp_id, emp_name, address, gender, position from tb_employee;
```

emp_id	emp_name	address	gender	position
501234	Andre1	jl. Merpati 53	F	Waitres
500500	Kurosaki ichigo	jl. Serenetai	M	Sales
505566	Lulu	jl. Ditempat 123	F	Admin

```
3 rows in set (0.00 sec)

mysql>
```

Gambar 6.10 Hasil Pengujian Registrasi Employee pada Database MySQL
Sumber : *Pengujian Dan Analisis*

6.3.2 Pengujian Proses Edit Data Employee

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data employee yang telah mengalami perubahan dapat disimpan dengan baik pada tabel tb_employee dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe|**.
- Memilih menu |**Edit Employee|** pada Employee Master Data.
- Memasukkan Nama Employee sebagai filter data.
- Menampilkan data Employee yang diinginkan.
- Melihat data employee pada database MySQL sebelum mengalami perubahan, dengan perintah :

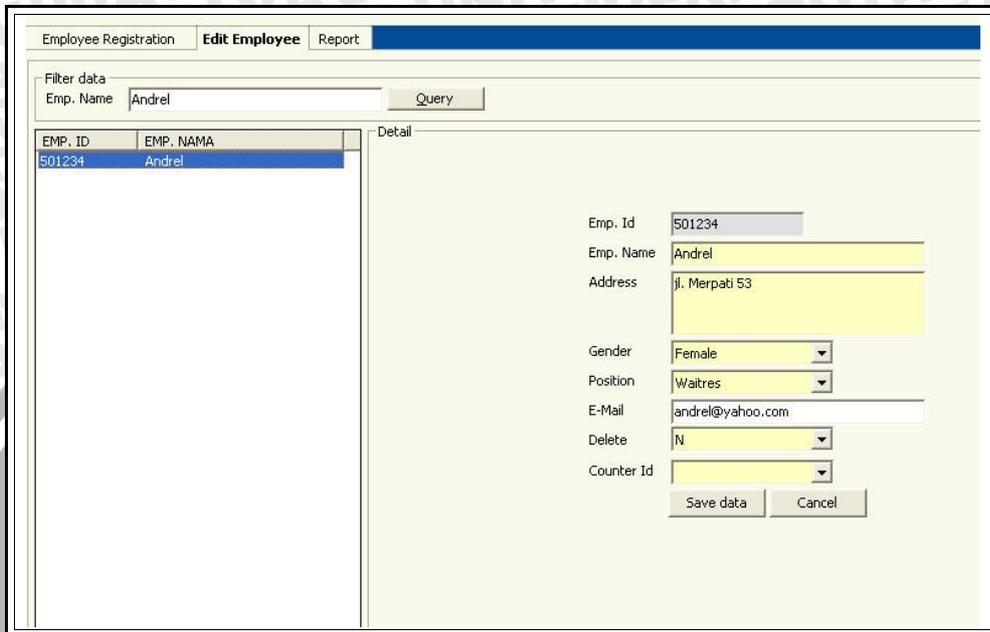
```
SELECT emp_name, address, email FROM tb_employee
WHERE emp_name = 'Andre1'
```

- Mengubah data Employee yang ditampilkan
- Melihat perubahan data Employee pada database MySQL dengan menggunakan perintah :

```
SELECT emp_name, address, position FROM tb_employee
WHERE emp_name = 'Andre1'
```

c. Hasil Pengujian dan Analisis

Antarmuka Edit Employee pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.11.



Gambar 6.11 Antarmuka Edit Employee
Sumber : *Pengujian Dan Analisis*

Data employee yang tersimpan pada database dapat diubah jika diinginkan pembaharuan/update data. Berikut Data Employee yang tersimpan pada database sebelum dilakukan update/edit data ditunjukkan pada Gambar 6.12

```
mysql> select emp_name, address, position from tb_employee where emp_name='andrel';
+-----+-----+-----+
| emp_name | address      | position |
+-----+-----+-----+
| Andrel   | jl. Merpati 53 | Waitres  |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Gambar 6.12 Data employee pada Database MySQL sebelum update data
Sumber : *Pengujian Dan Analisis*

Hasil update / edit data alamat employee yang telah dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran diperlihatkan dalam Gambar 6.13.

```
mysql> select emp_name, address, position from tb_employee where emp_name='andre
1';
+-----+-----+-----+
| emp_name | address          | position |
+-----+-----+-----+
| Andre1   | jl. Surabaya 32 | Waitres  |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Gambar 6.13 Hasil Pengujian Edit Employee pada Database MySQL

Sumber : *Pengujian Dan Analisis*

Berdasarkan tampilan diatas, tampak perubahan data alamat berhasil disimpan dengan baik pada tb_employee dalam database MySQL.

6.3.3 Pengujian Proses Report Data Employee

a. Tujuan

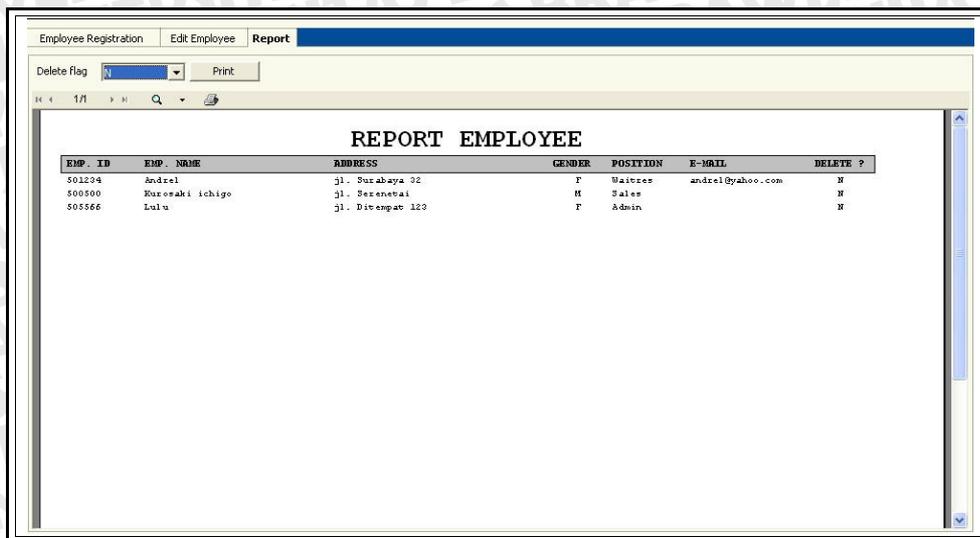
Pengujian ini dilakukan untuk melihat apakah Sistem Transaksi Pembayaran FoodCourt dapat melakukan proses menampilkan report data employee yang tersimpan didalam database foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** |.
- Memilih menu |**Report**| pada Employee Master Data.
- Menampilkan data employee pada database MySQL dengan perintah :
SELECT emp_id, emp_name, address, gender, position, email FROM tb_employee WHERE delete_flag ='N'.
- Menampilkan data employee pada database MySQL dengan perintah :
SELECT emp_id, emp_name, address, gender, position, email FROM tb_employee WHERE delete_flag ='Y'.

c. Hasil Pengujian dan Analisis

Antarmuka Report pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.14.



Gambar 6.14 Antarmuka Report Employee
 Sumber : Pengujian Dan Analisis

Report data employee diperoleh berdasarkan delete flag. Apabila status employee masih aktif maka delete flag-nya adalah N (No) begitu pula sebaliknya, status employee yang sudah tidak aktif maka delete flag-nya adalah Y (Yes). Berikut hasil query data employee aktif yang tersimpan pada tabel `tb_employee` dalam database `foodcourt` diperlihatkan pada Gambar 6.15.

```
mysql> select emp_id, emp_name, address, gender, position, email from tb_employee
e where delete_flag = 'N';
+-----+
| emp_id | emp_name      | address          | gender | position | email          |
+-----+
| 501234 | Andrei        | Jl. Surabaya 32 | F      | Waitres  | andre@yahoo.com |
| 500500 | Kurosaki ichigo | Jl. Serenetai   | M      | Sales    |                  |
| 505566 | Lulu          | Jl. Ditempat 123 | F      | Admin    |                  |
+-----+
3 rows in set (0.01 sec)
mysql>
```

Gambar 6.15 Hasil Pengujian Report Employee dengan Delete Flag "N/No"
 Sumber : Pengujian Dan Analisis

Sedangkan hasil query data employee non aktif yang tersimpan pada tabel `tb_employee` dalam database `foodcourt` diperlihatkan pada Gambar 6.16.

```
mysql> select emp_id, emp_name, address, gender, position, email from tb_employe
e where delete_flag
Empty_set (0.00 sec)
mysql>
```

Gambar 6.16 Hasil Pengujian Report Employee dengan Delete Flag "Y/Yes"

Sumber : *Pengujian Dan Analisis*

Berdasarkan tampilan diatas, dapat diketahui bahwa pada database foodcourt tidak terdapat data employee non aktif.

6.3.4 Pengujian Proses Registrasi Customer

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data customer dapat teregister dengan baik dan tersimpan pada tb_customer dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** |.
- Memilih menu customer pada Sistem Transaksi Pembayaran Food Court.
- Memilih menu |**Customer Registration**| pada Customer Master Data
- Memasukan data customer melalui Sistem Transaksi Pembayaran Food Court
- Menampilkan data customer setelah data disimpan pada tabel tb_customer dengan memberikan perintah :

```
SELECT cust_id, cust_name, address, credit_limit,
phone FROM tb_customer
```

c. Hasil Pengujian dan Analisis

Antarmuka Registrasi Customer pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.17:

CUST. ID	CUST. NAME	ADDRESS	PHONE	ID. NO.	GENDER	E-MAIL	DEL?	CREDIT
0809010002	Nesya	jl. Melati 77 Surabaya	085234390595	987654321	F		N	184000
0809010003	Redityo	jl. Teratai 196	058218610600	456123987	M		N	155000
0809010001	Vento	jl.Cempaka 33	0341458761	123456789	M	y@hoo.com	N	150000

Gambar 6.17 Antarmuka Registrasi Customer
 Sumber : *Pengujian Dan Analisis*

Data customer dimasukkan dengan memberikan parameter yang dibutuhkan pada Sistem Transaksi Pembayaran Food Court. Contoh data yang dimasukkan terdapat pada Tabel 6.2.

Tabel 6.2 Isi-Pengujian Data Customer

Cust. ID	Cust. Name	Address	Credit	Phone
0809010001	Vento	Jl.Cempaka 33	100000	0341458761
0809010002	Nesya	Jl. Kupu-kupu 53	150000	085234390595
0809010003	Redityo	Jl.Teratai 196	100000	085218610600

Hasil data yang dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran diperlihatkan dalam Gambar 6.18.

```
mysql> SELECT cust_id, cust_name, address, credit_limit, phone FROM tb_customer ;
+-----+-----+-----+-----+-----+
| cust_id | cust_name | address | credit_limit | phone |
+-----+-----+-----+-----+-----+
| 0809010001 | Vento | jl Cempaka 33 | 150000 | 0341458761 |
| 0809010002 | Nesya | jl. Melati 77 Surabaya | 184000 | 085234390595 |
| 0809010003 | Redityo | jl. Teratai 196 | 155000 | 058218610600 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Gambar 6.18 Hasil Pengujian Registrasi Customer pada Database MySQL
 Sumber : *Pengujian Dan Analisis*

6.3.5 Pengujian Proses Edit Data Customer

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data customer yang telah mengalami perubahan dapat disimpan dengan baik pada tabel `tb_customer` dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi**|`FoodCourtSys.exe`].
- Memilih menu |**Edit Customer**| pada Customer Master Data.
- Memasukkan Nama Customer sebagai filter data.
- Menampilkan data customer yang diinginkan.
- Melihat data customer pada database MySQL sebelum mengalami perubahan, dengan perintah :

```
SELECT cust_name, address, phone FROM tb_customer  
WHERE cust_name = 'Nesya';
```

- Mengubah data customer yaitu alamat customer
- Melihat perubahan data Customer pada database MySQL dengan menggunakan perintah :

```
SELECT cust_name, address, phone FROM tb_customer  
WHERE cust_name = 'Nesya';
```

c. Hasil Pengujian dan Analisis

Antarmuka Edit Customer pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.19.

Gambar 6.19 Antarmuka Edit Customer
 Sumber : *Pengujian Dan Analisis*

Data customer yang tersimpan pada database dapat diubah jika diinginkan pembaharuan/update data. Berikut Data customer yang tersimpan pada database sebelum dilakukan update/edit data ditunjukkan pada Gambar 6.20

```
mysql> select cust_name, address, phone from tb_customer where cust_name = 'Nesya';
+-----+-----+-----+
| cust_name | address          | phone      |
+-----+-----+-----+
| Nesya     | j1. Melati 77 Surabaya | 085234390595 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.20 Data alamat customer pada Database MySQL sebelum update data
 Sumber : *Pengujian Dan Analisis*

Hasil update data customer berupa alamat yang telah dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran diperlihatkan dalam Gambar 6.21.

```
mysql> select cust_name, address, phone from tb_customer where cust_name = 'Nesya';
+-----+-----+-----+
| cust_name | address          | phone      |
+-----+-----+-----+
| Nesya     | j1. Kupa kupa 53 Surabaya | 085234390595 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.21 Hasil Pengujian Edit Data alamat Customer pada Database MySQL
 Sumber : *Pengujian Dan Analisis*

Berdasarkan Gambar 6.21, tampak bahwa perubahan data alamat berhasil



disimpan dengan baik pada `tb_customer` dalam database MySQL.

6.3.6 Pengujian Proses Report Data Customer

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah Sistem Transaksi Pembayaran FoodCourt dapat melakukan proses menampilkan report data customer yang tersimpan didalam database `foodcourt`.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada |**Path Aplikasi|FoodCourtSys.exe**|.
- Memilih menu |**Report**| pada Customer Master Data.
- Menampilkan data Customer pada database MySQL dengan perintah:
`SELECT cust_id, cust_name, address, gender, phone, email FROM tb_customer WHERE delete_flag = 'N'`.
- Menampilkan data customer pada database MySQL dengan perintah:
`SELECT cust_id, cust_name, address, gender, phone, email FROM tb_customer WHERE delete_flag = 'Y'`.

c. Hasil Pengujian dan Analisis

Antarmuka Report pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.22.

CUST_ID	CUST_NAME	ADDRESS	GENDER	ID NO.	PHONE	DEL?	E-MAIL
0009010002	Hevya	Jl. Ngopo-bupyo 53 Surabaya	F	907654721	055234390595	N	
0009010003	Redityo	Jl. Teratai 126	M	456123907	0562126120600	N	
0009010001	Vento	Jl. Cengakha 23	M	123456789	0241450761	N	y@oo.com

Gambar 6.22 Antarmuka Report Customer
Sumber : *Pengujian Dan Analisis*

Report data customer diperoleh berdasarkan delete flag. Apabila status customer masih terdaftar dan aktif maka delete flag-nya adalah N (No) begitu pula sebaliknya, status customer yang sudah tidak aktif maka delete flag-nya adalah Y (Yes). Berikut hasil query data customer aktif yang tersimpan pada tabel tb_customer dalam database foodcourt diperlihatkan pada Gambar 6.23.

```
mysql> SELECT cust_id, cust_name, address, gender, phone, email
-> FROM tb_customer WHERE delete_flag = 'N';
```

cust_id	cust_name	address	gender	phone	email
0809010001	Vento	jl Cempaka 33	M	0341458761	y@hoo.com
0809010002	Nesya	jl. Kupu-kupu 53 Surabaya	F	085234390595	
0809010003	Redityo	jl. Teratai 196	M	058218610600	

```
3 rows in set (0.00 sec)
```

Gambar 6.23 Hasil Pengujian Report Customer dengan Delete Flag "N/No"
Sumber : *Pengujian Dan Analisis*

Sedangkan hasil query data customer non aktif yang tersimpan pada tabel tb_customer dalam database foodcourt diperlihatkan pada Gambar 6.24.

```
mysql> select cust_id, cust_name, address, gender, phone, email from tb_customer
where delete_flag='Y';
```

```
Empty set (0.00 sec)
```

```
mysql> -
```

Gambar 6.24 Hasil Pengujian Report Customer dengan Delete Flag "Y/Yes"
Sumber : *Pengujian Dan Analisis*

Berdasarkan tampilan tersebut, dapat diketahui bahwa pada database foodcourt tidak terdapat data customer yang sudah tidak terdaftar atau non aktif.

6.3.7 Pengujian Proses Penambahan Kredit pada E-card.

a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah proses penambahan kredit pada E-card dapat dilakukan dengan baik dan dapat tersimpan pada tabel tb_customer dalam database foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada |



Path Aplikasi|FoodCourtSys.exe|.

- Memilih menu |**Edit Customer**| pada Employee Master Data.
- Memasukkan Nama Customer sebagai filter data.
- Memilih link |**Add Credit**| pada details data customer.
- Menampilkan data Customer pada database MySQL sebelum dilakukan penambahan kredit dengan perintah:

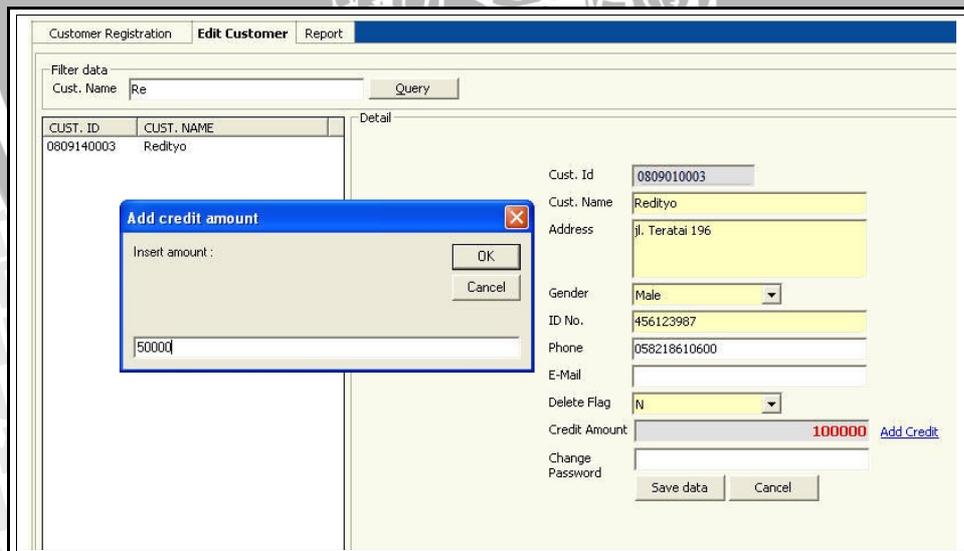
```
SELECT credit_limit FROM tb_customer WHERE cust_name ='Redityo'.
```

- Memasukkan nominal yang akan ditambahkan.
- Menampilkan data Customer pada database MySQL setelah dilakukan penambahan kredit dengan perintah:

```
SELECT credit_limit FROM tb_customer WHERE cust_name ='Redityo'.
```

c. Hasil Pengujian dan Analisis

Antarmuka Report pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.25.



Gambar 6.25 Antarmuka Add Credit
Sumber : Pengujian Dan Analisis



Pada proses penambahan kredit, Customer harus membayar sejumlah uang sesuai dengan nominal kredit yang akan ditambahkan pada E-card. Kemudian sistem akan memproses dan menambahkan dengan sisa kredit yang ada sebelumnya. Hasil Query dari proses penambahan kredit diperlihatkan dalam Gambar 6.26.

```
mysql> select credit_limit from tb_customer where cust_name='redityo';
+-----+
| credit_limit |
+-----+
|      150000  |
+-----+
1 row in set (0.05 sec)

mysql> _
```

Gambar 6.26 Hasil Pengujian Proses Penambahan Kredit
Sumber : Pengujian Dan Analisis

6.3.8 Pengujian Proses Registrasi Counter Makanan

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data counter makanan dapat teregister dengan baik dan tersimpan pada `tb_employee` dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** |.
- Memilih menu *counter/shop* pada Sistem Transaksi Pembayaran Food Court.
- Memilih menu |**Counter Registration**| pada Counter/Shop Master Data.
- Memasukan data *counter/shop* melalui Sistem Transaksi Pembayaran Food Court.
- Menampilkan data counter setelah data disimpan pada tabel `tb_counter` dengan memberikan perintah :

```
SELECT counter_id, counter_name, address,
```

company_name, share_amt FROM tb_counter.

c. Hasil Pengujian dan Analisis

Antarmuka Registrasi Employee pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.27:

counter_id	counter name	address	phone	company name	share_amt
S02	Big Burger	jl. Pelan 66	8848894	PT. Semua Suka	10
S03	Fiesta Steak	jl. Kelinci 32	327369	Fiesta	15
S01	Kebab Ali Baba	jl. Maju 3	774411	Ali Baba Corp.	10

Gambar 6.27 Antarmuka Registrasi Counter Makanan
Sumber : *Pengujian Dan Analisis*

Counter ID	Counter Name	Address	Company Name	Share Amount (%)
S01	Kebab Ali Baba	Jl. Maju 3	Ali Baba Corp.	10
S02	Big Burger	Jl. Pelan 66	PT. Manasuka	10
S03	Fiesta Steak	Jl. Kelinci 32	Fiesta	15

Data counter dimasukkan dengan memberikan parameter yang dibutuhkan pada Sistem Transaksi Pembayaran Food Court. Contoh data yang dimasukkan terdapat pada Tabel 6.3.

Tabel 6.3 Isi Pengujian Data Counter

Hasil data yang dimasukkan pada database MySQL melalui Sistem



Transaksi Pembayaran diperlihatkan dalam Gambar 6.28.

```
mysql> select counter_id, counter_name, address, company_name, share_amt from tb_counter;
```

counter_id	counter_name	address	company_name	share_amt
S01	Kebab Ali Baba	Jl. Maju 3	Ali Baba Corp.	10
S02	Big Burger	Jl. Pelan 66	PT. Semua Suka	10
S03	Fiesta Steak	Jl. Kelinci 32	Fiesta	15

```
3 rows in set (0.00 sec)
```

```
mysql>
```

Gambar 6.28 Hasil Pengujian Registrasi Counter pada Database MySQL
Sumber : *Pengujian Dan Analisis*

6.3.9 Pengujian Proses Edit Data Counter Makanan

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data counter yang telah mengalami perubahan dapat disimpan dengan baik pada tabel `tb_counter` dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe|**.
- Memilih menu |**Edit Counter**| pada Counter/Shop Master Data.
- Memasukkan Nama Counter sebagai filter data.
- Menampilkan data counter yang diinginkan.
- Melihat data counter pada database MySQL sebelum mengalami perubahan, dengan perintah :

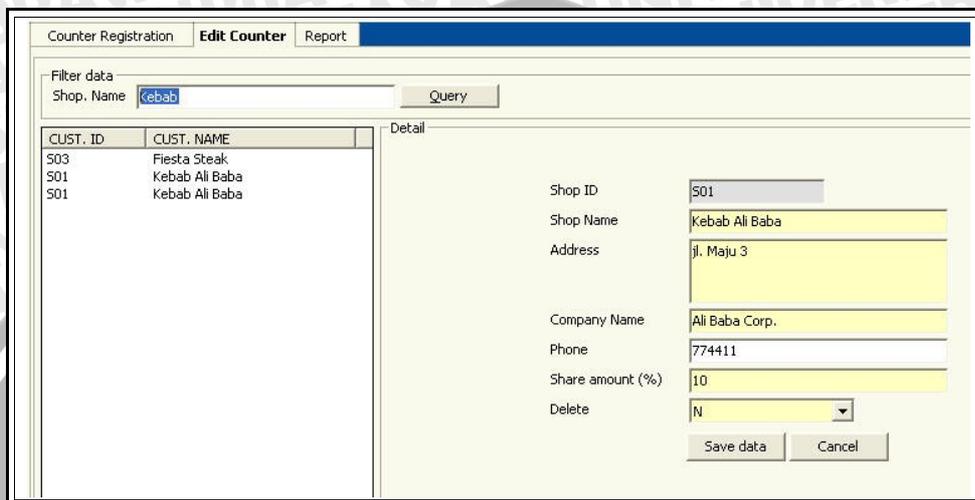
```
SELECT address, company_name, share_amt FROM  
tb_counter WHERE counter_name = 'Kebab Ali Baba'
```

- Mengubah data counter yaitu besarnya prosentase bagi hasil (share amount) yang dibebankan kepada counter makanan.
- Melihat perubahan data counter pada database MySQL dengan menggunakan perintah :

```
SELECT address, company_name, share_amt FROM  
tb_counter WHERE counter_name = 'Kebab Ali Baba'
```

c. Hasil Pengujian dan Analisis

Antarmuka Edit Counter pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.29.



Gambar 6.29 Antarmuka Edit Counter
 Sumber : *Pengujian Dan Analisis*

Data counter yang tersimpan pada database dapat diubah jika diinginkan pembaharuan/update data. Berikut Data counter yang tersimpan pada database sebelum dilakukan update/edit data ditunjukkan pada Gambar 6.30.

```
mysql> select address, company_name, share_ant from tb_counter where counter_name='kebab ali baba';
+-----+-----+-----+
| address | company_name | share_ant |
+-----+-----+-----+
| j1. Maju 3 | Ali Baba Corp. | 10 |
+-----+-----+-----+
1 row in set (0.03 sec)

mysql>
```

Gambar 6.30 Data counter pada Database MySQL sebelum update data
 Sumber : *Pengujian Dan Analisis*

Hasil update data customer berupa prosentase bagi hasil / share amount yang telah dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran diperlihatkan dalam Gambar 6.31.

```
mysql> select address, company_name, share_ant from tb_counter where counter_name='kebab ali baba';
+-----+-----+-----+
| address | company_name | share_ant |
+-----+-----+-----+
| j1. Maju 3 | Ali Baba Corp. | 15 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Gambar 6.31 Hasil Pengujian Edit Data Customer pada Database MySQL
 Sumber : *Pengujian Dan Analisis*

Berdasarkan Gambar 6.31, tampak bahwa perubahan prosentase bagi hasil/*share amount* berhasil disimpan dengan baik pada *tb_customer* dalam database MySQL.

6.3.10 Pengujian Proses Report Data Counter Makanan

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah Sistem Transaksi Pembayaran FoodCourt dapat melakukan proses menampilkan report data counter yang tersimpan didalam database *foodcourt*.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada **[Path Aplikasi|FoodCourtSys.exe]**.
- Memilih menu **[Report]** pada Counter/Shop Master Data.
- Menampilkan data Counter pada database MySQL dengan perintah:

```
SELECT counter_id, counter_name, address,
company_name, phone, share_amt FROM tb_counter
WHERE delete_flag = 'N'.
```
- Menampilkan data employee pada database MySQL dengan perintah:

```
SELECT counter_id, counter_name, address,
company_name, phone, share_amt FROM tb_counter
WHERE delete_flag = 'Y'.
```

c. Hasil Pengujian dan Analisis

Antarmuka Report pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.32.

SHOP ID	COUNTER/SHOP NAME	ADDRESS	COMPANY NAME	PHONE	SHARE (%)	DEL?
S02	Big Burger	Jl. Pelan 66	PT. Semua Suka	8848894	10	N
S03	Fiesta Steak	Jl. Kelinci 32	Fiesta	327369	15	N
S01	Kebab Ali Baba	Jl. Maju 3	Ali Baba Corp.	774411	15	N

Gambar 6.32 Antarmuka Report Counter
Sumber : *Pengujian Dan Analisis*

Report data counter diperoleh berdasarkan delete flag. Apabila status counter aktif maka delete flag-nya adalah N (No) begitu pula sebaliknya, status counter yang sudah tidak aktif maka delete flag-nya adalah Y (Yes). Berikut hasil query data counter aktif yang tersimpan pada tabel `tb_counter` dalam database `foodcourt` diperlihatkan pada Gambar 6.33.

```
mysql> select counter_id, counter_name, address, company_name, phone, share_ant from tb_counter
-> where delete_flag='N';
```

counter_id	counter_name	address	company_name	phone	share_ant
S01	Kebab Ali Baba	Jl. Maju 3	Ali Baba Corp.	774411	15
S02	Big Burger	Jl. Pelan 66	PT. Semua Suka	8848894	10
S03	Fiesta Steak	Jl. Kelinci 32	Fiesta	327369	15

```
3 rows in set (0.00 sec)

mysql>
```

Gambar 6.33 Hasil Pengujian Report Counter dengan Delete Flag "N/No"
Sumber : *Pengujian Dan Analisis*

Sedangkan hasil query data counter non aktif yang tersimpan pada tabel `tb_counter` dalam database `foodcourt` diperlihatkan pada Gambar 6.34.

```
mysql> select counter_id, counter_name, address, company_name, phone, share_ant from tb_counter
-> where delete_flag='Y';
Empty set (0.00 sec)

mysql>
```

Gambar 6.34 Hasil Pengujian Report Counter dengan Delete Flag "Y/Yes"
Sumber : *Pengujian Dan Analisis*

Berdasarkan tampilan diatas, dapat diketahui bahwa pada database `foodcourt` tidak terdapat data counter yang sudah tidak terdaftar atau non aktif.

6.3.11 Pengujian Report Transaksi Harian



a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah sistem dapat melakukan proses menampilkan *report* transaksi harian yang terjadi pada setiap *counter* makanan yang tersimpan dalam *database* foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi**|FoodCourtSys.exe| .
- Memilih menu |**Transaction by Shop – Daily**| pada Report.
- Pilih tanggal transaksi dan nama counter makanan, dimana transaksi terjadi.
- Menampilkan report transaksi harian pada counter makanan “Fiesta Steak” yang terdapat pada database MySQL dengan memberikan perintah :

```
SELECT mst.date trans_date,mst.trans_id,
       ctr.counter_name,cust.cust_name,food.food_name,
       dtl.qty,dtl.price,dtl.status_order,
       (dtl.price * dtl.qty) sub_total
FROM tb_trans_mstr mst,tb_trans_dtl dtl,
     tb_counter ctr,tb_food food,tb_customer cust,
     tb_employee emp
WHERE mst.trans_id = dtl.trans_id
      and mst.counter_id = ctr.counter_id
      and dtl.food_id = food.food_id
      and mst.customer_id = cust.cust_id
      and mst.emp_id = emp.emp_id
      and mst.cancel_flag = 'N'
      and dtl.status_order in ('NEW','DELIVER')
      and mst.counter_id = 's03';
```

c. Hasil Pengujian dan Analisis

Antarmuka Report Transaksi Harian pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.35:

Transaction By Shop - Daily | Transaction By Shop - Monthly | Share By Shop | Customer transaction

Date: 20/11/2008 ~ 20/11/2008 Shop: S03 * Fiesta Steak

REPORT TRANSACTION BY SHOP (DAILY)

COUNTER NAME	TRANS. ID	TRANS. DATE	CUSTOMER	FOOD NAME	PRICE	DISC	QTY	STATUS	SUB TOTAL
Fiesta Steak	11	20/11/2008	Nesya	STEAK TENDERLOIN	25.000	0	1	DELIVER	25.000
								SUB :	25.000
	11	20/11/2008	Nesya	ORANGE JUICE	5.000	0	1	DELIVER	5.000
								SUB :	5.000
								TOTAL :	30.000
								GRAND TOTAL :	30.000

Gambar 6.35 Antarmuka Report Transaksi Harian
Sumber : Pengujian Dan Analisis

Report transaksi harian didapat berdasarkan transaksi yang terjadi pada setiap counter makanan, untuk mengetahui data transaksi yang terjadi pada counter makanan “Fiesta Steak”, maka nama counter dapat dicari pada list shop. Hasil query data transaksi harian pada counter makanan “Fiesta Steak” diperlihatkan dalam Gambar 6.36.

```
mysql> SELECT mst.date trans_date,mst.trans_id,
-> ctr.counter_name,cust.cust_name,food.food_name,dtl.qty,dtl.price,dtl.stat
us_order,
-> (dtl.price * dtl.qty) sub_total
-> FROM tb_trans_mstr mst,tb_trans_dtl dtl,
-> tb_counter ctr,tb_food food,tb_customer cust, tb_employee emp
-> WHERE mst.trans_id = dtl.trans_id
-> and mst.counter_id = ctr.counter_id
-> and dtl.food_id = food.food_id
-> and mst.customer_id = cust.cust_id
-> and mst.emp_id = emp.emp_id
-> and mst.cancel_flag = 'N'
-> and dtl.status_order in ('NEW','DELIUER')
-> and mst.counter_id = 's03';
```

trans_date	trans_id	counter_name	cust_name	food_name	qty	price	status_order	sub_total
2008-11-20 11:13:03	11	Fiesta Steak	Nesya	STEAK TENDERLOIN	1	25000	DELIUER	25000
2008-11-20 11:13:03	11	Fiesta Steak	Nesya	ORANGE JUICE	1	5000	DELIUER	5000

2 rows in set (0.00 sec)

Gambar 6.36 Hasil Pengujian Report Transaksi Harian
Sumber : Pengujian Dan Analisis

6.3.12 Pengujian Report Transaksi Bulanan

a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah sistem dapat melakukan proses menampilkan report transaksi bulanan yang terjadi pada setiap *counter* makanan yang tersimpan dalam *database* foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi**|FoodCourtSys.exe| .
- Memilih menu |**Transaction by Shop – Monthly**| pada Report.
- Masukkan bulan dan tahun transaksi dengan format YYYYMM dan pilih nama counter makanan, dimana transaksi terjadi.
- Menampilkan report transaksi bulanan pada counter makanan “Kebab Alibaba” yang terdapat pada database MySQL dengan memberikan perintah :

```
SELECT mst.date trans_date,mst.trans_id,
       ctr.counter_name,cust.cust_name,food.food_name,
       dtl.qty,dtl.price,dtl.status_order,
       (dtl.price * dtl.qty) sub_total
FROM tb_trans_mstr mst,tb_trans_dtl dtl,
     tb_counter ctr,tb_food food,tb_customer cust,
     tb_employee emp
WHERE mst.trans_id = dtl.trans_id
      and mst.counter_id = ctr.counter_id
      and dtl.food_id = food.food_id
      and mst.customer_id = cust.cust_id
      and mst.emp_id = emp.emp_id
      and mst.cancel_flag = 'N'
      and dtl.status_order in ('NEW','DELIVER')
      and mst.counter_id = 's01';
```

c. Hasil Pengujian dan Analisis

Antarmuka Report Transaksi Bulanan pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.37:

Transaction By Shop - Daily **Transaction By Shop - Monthly** Share By Shop Customer transaction

Month: 200811 Shop: S01 * Kebab Ali Baba Print

1/1

REPORT TRANSACTION BY SHOP (MONTHLY)

COUNTER NAME	TRANS. DATE	FOOD NAME	PRICE	DISC	QTY	STATUS	AMOUNT
Kebab Ali Baba							
1	20/11/2008	KEBAP SINDANG	10.000	0	1	DELIVER	10.000
							AMOUNT : 10.000
2	20/11/2008	KEBAP SINDANG	4.000	0	2	DELIVER	8.000
							AMOUNT : 8.000
3	21/11/2008	KEBAP SINDANG	15.000	0	1	DELIVER	15.000
							AMOUNT : 15.000
4	21/11/2008	KEBAP SINDANG	4.000	0	1	DELIVER	4.000
							AMOUNT : 4.000
5	21/11/2008	KEBAP SINDANG	10.000	0	2	DELIVER	20.000
							AMOUNT : 20.000
6	21/11/2008	KEBAP SINDANG	10.000	0	2	DELIVER	20.000
							AMOUNT : 20.000
7	21/11/2008	KEBAP SINDANG	4.000	0	2	DELIVER	8.000
							AMOUNT : 8.000
GRAND TOTAL :							75.000

Gambar 6.37 Antarmuka Report Transaksi Bulanan
 Sumber : *Pengujian Dan Analisis*

Report transaksi bulanan didapat berdasarkan transaksi harian yang terjadi pada setiap counter makanan berdasarkan pada bulan ketika transaksi tersebut terjadi. Dimana pada report bulanan, pihak counter juga dapat mengetahui besarnya total pendapatan yang diperoleh selama sebulan. Untuk mengetahui keseluruhan data transaksi dan total pendapatan yang terjadi pada counter makanan “Kebab Ali Baba” selama satu bulan, maka nama counter dapat dicari pada list shop. Hasil query data transaksi bulanan pada counter makanan “Kebab Ali Baba” diperlihatkan dalam Gambar 6.38.

```
mysql> SELECT mst.date trans_date,mst.trans_id,
-> ctr.counter_name,cust.cust_name,food.food_name,dtl.qty,dtl.price,dtl.stat
us_order,
-> (dtl.price * dtl.qty) sub_total
-> FROM tb_trans_mstr mst,tb_trans_dtl dtl,
-> tb_counter ctr,tb_food food,tb_customer cust, tb_employee emp
-> WHERE mst.trans_id = dtl.trans_id
-> and mst.counter_id = ctr.counter_id
-> and dtl.food_id = food.food_id
-> and mst.customer_id = cust.cust_id
-> and mst.emp_id = emp.emp_id
-> and mst.cancel_flag = 'N'
-> and dtl.status_order in ('NEW','DELIVER')
-> and mst.counter_id = 's01';
```

trans_date	trans_id	counter_name	cust_name	food_name	q
ty	price	status_order	sub_total		
2008-11-21 18:23:41	5	Kebab Ali Baba	Redityo	KEBAB SUPER	1
1	15000	DELIVER			15000
2008-11-21 18:23:41	5	Kebab Ali Baba	Redityo	TEH TARIK	1
1	4000	DELIVER			4000
2008-11-20 20:14:56	1	Kebab Ali Baba	Vento	KEBAB SEDANG	1
1	10000	DELIVER			10000
2008-11-20 20:15:36	2	Kebab Ali Baba	Nesya	TEH TARIK	1
2	4000	DELIVER			8000
2008-11-21 18:26:01	6	Kebab Ali Baba	Nesya	ROTI CANE	1
2	5000	DELIVER			10000
2008-11-21 18:26:01	6	Kebab Ali Baba	Nesya	KEBAB SEDANG	1
2	10000	DELIVER			20000
2008-11-21 18:26:01	6	Kebab Ali Baba	Nesya	TEH TARIK	1
2	4000	DELIVER			8000

7 rows in set (0.47 sec)

Gambar 6.38 Hasil Pengujian Report Transaksi Bulanan
 Sumber: Pengujian Dan Analisis

6.3.13 Pengujian Report Bagi Hasil

a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah Sistem dapat melakukan proses menampilkan report bagi hasil dari setiap counter makanan yang tersimpan dalam database foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe|** .
- Memilih menu |**Share by Shop**| pada Report.
- Masukkan bulan dan tahun dengan format YYYYMM.
- Menampilkan report bagi hasil pada database MySQL dengan memberikan perintah :

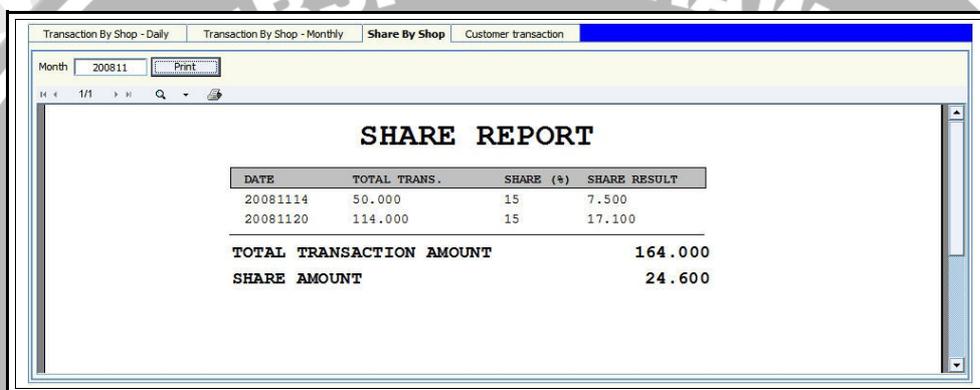
```
SELECT date_format(mst.date,'%Y%m%d') trans_date,
sum((dtl.price * dtl.qty) - ((dtl.price *
dtl.qty) * dtl.disc/100))sub_total,ctr.share_amt,
sum((dtl.price * dtl.qty) - ((dtl.price *
```

```

        dtl.qty) * dtl.disc/100)) * (ctr.share_amt / 100)
    share_result
FROM    tb_trans_mstr    mst,    tb_trans_dtl    dtl,
        tb_counter ctr,tb_food food
WHERE   mst.trans_id = dtl.trans_id
        and mst.counter_id = ctr.counter_id
        and mst.cancel_flag = 'N'
        and dtl.status_order = 'DELIVER'
        and dtl.food_id = food.food_id
GROUP BY trans_date
ORDER BY mst.date asc;
    
```

c. Hasil Pengujian dan Analisis

Antarmuka Report Transaksi Bulanan pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.39:



Gambar 6.39 Antarmuka Report Bagi Hasil
Sumber : *Pengujian Dan Analisis*

Report bagi hasil diperoleh berdasarkan transaksi harian yang terjadi pada setiap counter makanan. Dimana pada report bagi hasil ini, pihak counter dapat mengetahui besarnya total pendapatan yang diperoleh selama sebulan dan besarnya pembagian pendapatan berdasarkan prosentase bagi hasil yang sudah disepakati sebelumnya. Berikut hasil query report bagi hasil yang diperlihatkan dalam Gambar 6.40.



```
mysql> select date_format(mst.date,'%Y%m%d') trans_date,
-> sum((dtl.price * dtl.qty) - ((dtl.price * dtl.qty) * dtl.disc/100)) sub
total,ctr.share_amt,
-> sum((dtl.price * dtl.qty) - ((dtl.price * dtl.qty) * dtl.disc/100)) * (
ctr.share_amt / 100) share_result
-> from tb_trans_mstr mst, tb_trans_dtl dtl, tb_counter ctr,tb_food food
-> where mst.trans_id = dtl.trans_id
-> and mst.counter_id = ctr.counter_id
-> and mst.cancel_flag = 'N'
-> and dtl.status_order = 'DELIVER'
-> and dtl.food_id = food.food_id
-> group by trans_date
-> order by mst.date asc
-> ;
```

trans_date	sub_total	share_amt	share_result
20080920	150000	15	22500
20081114	50000	15	7500
20081120	81000	15	12150

3 rows in set (0.00 sec)

Gambar 6.40 Hasil Pengujian Report Bagi Hasil
Sumber : *Pengujian Dan Analisis*

6.3.14 Pengujian Report Transaksi Customer

a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah Sistem dapat melakukan proses menampilkan report bagi hasil dari setiap counter makanan yang tersimpan dalam database foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** | .
- Memilih menu |**Share by Shop**| pada Report.
- Masukkan bulan dan tahun dengan format YYYYMM.
- Menampilkan report bagi hasil pada database MySQL dengan memberikan perintah :

```
SELECT trans_date,trans_type,cust_id,cust_name,
counter_name,amount
FROM tb_trans_hist a,tb_counter b,tb_customer c
WHERE a.customer_id = c.cust_id
and a.counter_id = b.counter_id
and a.customer_id = '0809010002'
```

c. Hasil Pengujian dan Analisis

Antarmuka Report Transaksi Customer pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.41

DATE/TIME	TRANSACTION	COUNTER NAME	AMOUNT
20/11/2008 20:15:PAYMENT	Kebab Ali Baba		8.000,00
20/11/2008 20:16:PAYMENT	Big Burger		8.000,00
21/11/2008 18:26:PAYMENT	Kebab Ali Baba		38.000,00
TOTAL TRANSACTION			54.000,00

Gambar 6.41 Antarmuka Report Transaksi Customer
Sumber : *Pengujian Dan Analisis*

Report transaksi customer diperoleh berdasarkan transaksi yang dilakukan oleh customer pada setiap counter makanan. Dimana pada report transaksi customer ini, customer dapat mengetahui besarnya transaksi yang telah dilakukan selama sebulan. Berikut hasil query report bagi hasil yang diperlihatkan dalam Gambar 6.42.

```
mysql> select trans_date, trans_type, cust_id, cust_name, counter_name, amount
-> from tb_trans_hist a, tb_counter b, tb_customer c
-> where a.customer_id = c.cust_id
-> and a.counter_id = b.counter_id
-> and a.customer_id = '0809010002';
```

trans_date	trans_type	cust_id	cust_name	counter_name	amount
2008-11-20 20:15:36	PAYMENT	0809010002	Nesya	Kebab Ali Baba	8000
2008-11-21 18:26:01	PAYMENT	0809010002	Nesya	Kebab Ali Baba	38000
2008-11-20 20:16:23	PAYMENT	0809010002	Nesya	Big Burger	8000

3 rows in set (0.00 sec)

Gambar 6.42 Hasil Pengujian Report Transaksi Customer
Sumber : *Pengujian Dan Analisis*

6.3.15 Pengujian Proses User Management

a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah Sistem dapat membentuk hak user dalam penggunaan Sistem Transaksi Pembayaran Food Court dan data user dapat tersimpan pada `tb_user` dalam database MySQL dengan baik.

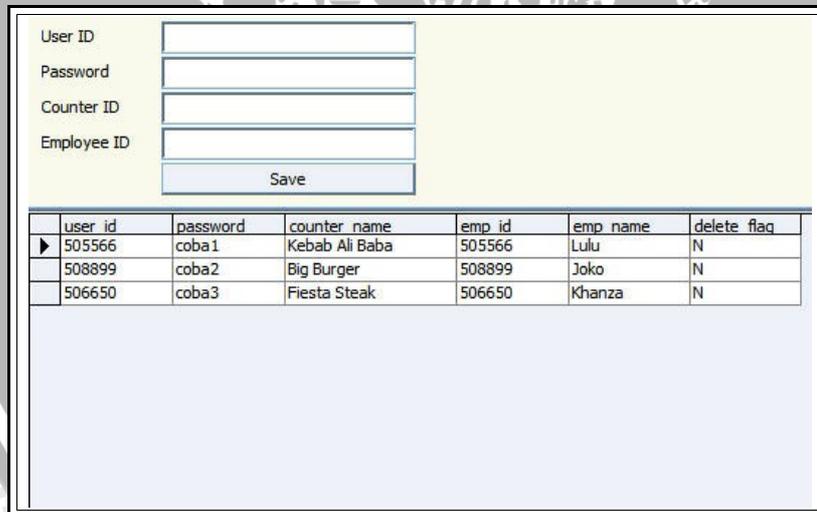
b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** | .
- Memilih menu User Management pada Sistem Transaksi Pembayaran Food Court.
- Masukkan data user melalui Sistem Transaksi Pembayaran Food Court.
- Menampilkan data user setelah data disimpan pada tabel `tb_user` dengan memberikan perintah :

```
SELECT user_id, password, counter_id, emp_id FROM
tb_user GROUP BY user_id asc;
```

c. Hasil Pengujian dan Analisis

Antarmuka User Management pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.43:



Gambar 6.43 Antarmuka User Management
Sumber : *Pengujian Dan Analisis*

Data user dimasukkan dengan memberikan parameter yang dibutuhkan pada Sistem Transaksi Pembayaran Food Court. Contoh Data yang dimasukkan terdapat pada Tabel 6.4

Tabel 6.4 Isi Pengujian Data User



User ID	Password	Counter ID	Employee ID
505566	coba1	S01	505566
508899	coba2	S02	508899
506650	coba3	S03	506650

Hasil

data yang dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran Food Court diperlihatkan dalam Gambar 6.44.

```
mysql> SELECT user_id, password, counter_id, emp_id
-> FROM tb_user GROUP BY user_id asc;
+-----+-----+-----+-----+
| user_id | password | counter_id | emp_id |
+-----+-----+-----+-----+
| 505566  | coba1   | s01        | 505566 |
| 506650  | coba3   | s03        | 506650 |
| 508899  | coba2   | s02        | 508899 |
| sysadmin | test    | admin      | sysadmin |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Gambar 6.44 Hasil Pengujian Input data User
Sumber : Pengujian Dan Analisis

Pada hasil query diatas, tampak bahwa terdapat user id ='sysadmin', user tersebut tidak ditampilkan dalam list data user pada Halaman User Management karena user melakukan login sebagai sysadmin.

6.4 Pengujian Aplikasi Operator Counter

6.4.1 Pengujian Proses Registrasi Data Food

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data food / item dapat teregister dengan baik dan tersimpan pada tb_food dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | Path Aplikasi|FoodCourtSys.exe|.
- Memilih menu food/item pada Sistem Transaksi Pembayaran Food Court.

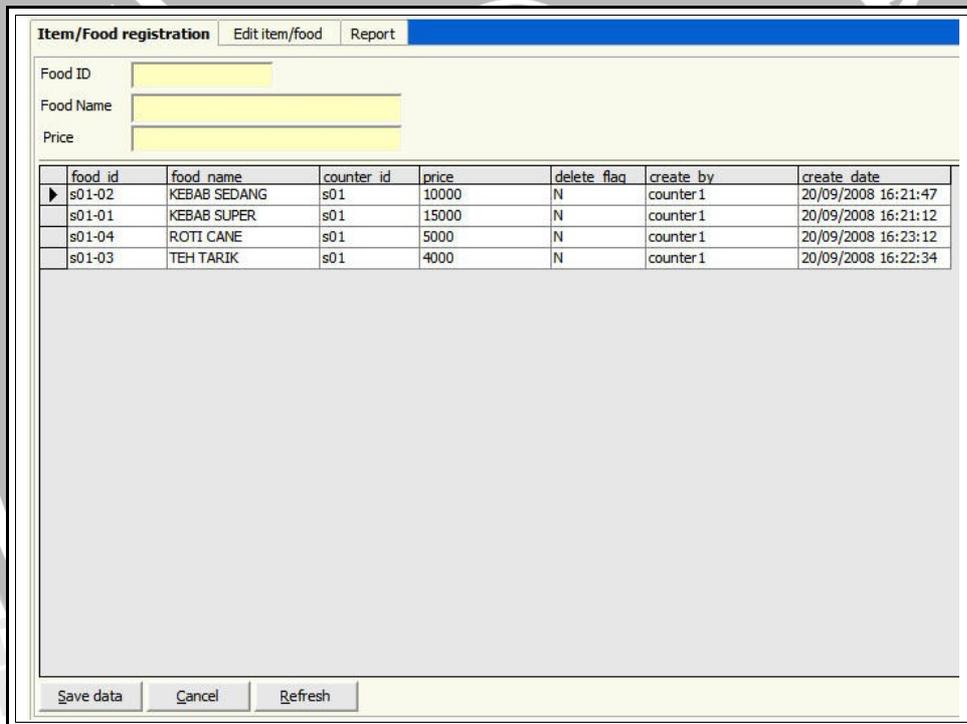


- Memilih menu **|Food Registration|** pada Food Master Data
- Memasukan data food melalui Sistem Transaksi Pembayaran Food Court
- Menampilkan data food setelah data disimpan pada tabel `tb_food` dengan memberikan perintah :

```
SELECT food_id, food_name, price, FROM tb_food
WHERE counter_id = 's01';
```

c. Hasil Pengujian dan Analisis

Antarmuka Registrasi Food/Item pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.45:



Gambar 6.45 Antarmuka Registrasi Food/Item
 Sumber : *Pengujian Dan Analisis*

Data food/item dimasukkan dengan memberikan parameter yang dibutuhkan pada Sistem Transaksi Pembayaran Food Court. Contoh data yang dimasukkan terdapat pada Tabel 6.5.

Tabel 6.5 Isi Pengujian Data Food/Item



Food ID	Food Name	Price
01	Kebab Super	15000
02	Kebab Sedang	10000
03	Teh Tarik	4000
04	Roti Cane	5000

ID food yang akan disimpan

pada tb_food akan ditambahkan dengan ID counter secara otomatis oleh sistem, hasil data yang dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran Food Court diperlihatkan dalam Gambar 6.46.

```
mysql> select food_id, food_name, price from tb_food where counter_id = 's01';
+-----+-----+-----+
| food_id | food_name | price |
+-----+-----+-----+
| s01-04 | ROTI CANE | 5000  |
| s01-03 | TEH TARIK | 4000  |
| s01-02 | KEBAB SEDANG | 10000 |
| s01-01 | KEBAB SUPER | 15000 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Gambar 6.46 Hasil Pengujian Registrasi Food/Item pada Database MySQL
Sumber : Pengujian Dan Analisis

6.4.2 Pengujian Proses Edit Data Food

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah data customer yang telah mengalami perubahan dapat disimpan dengan baik pada tabel tb_food dalam database MySQL.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | Path Aplikasi|FoodCourtSys.exe|.
- Memilih menu |Edit Food/Item| pada Food Master Data.
- Memasukkan Nama food sebagai filter data.
- Menampilkan data food yang diinginkan.
- Melihat data food pada database MySQL sebelum mengalami



perubahan, dengan perintah :

```
SELECT food_id, food_name, price FROM tb_food WHERE
food_name = 'Roti Cane';
```

- Mengubah data food yaitu harga dari makanan yang dijual.
- Melihat perubahan data food pada database MySQL dengan menggunakan perintah :

```
SELECT food_id, food_name, price FROM tb_food WHERE
food_name = 'Roti Cane';
```

c. Hasil Pengujian dan Analisis

Antarmuka Edit Food pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.47.

CUST. ID	CUST. NAME
s01-04	ROTI CANE

Detail

Food ID: s01-04
 Food Name: ROTI CANE
 Price: 5000
 Delete: N

Save data Cancel

Gambar 6.47 Antarmuka Edit Food
 Sumber : *Pengujian Dan Analisis*

Data food yang tersimpan pada database dapat diubah jika diinginkan pembaharuan/update data. Berikut Data food yang tersimpan pada database sebelum dilakukan update/edit data ditunjukkan dalam Gambar 6.48

```
mysql> select food_id, food_name, price from tb_food
-> where food_name=' Roti Cane';
+-----+-----+-----+
| food_id | food_name | price |
+-----+-----+-----+
| s01-04 | ROTI CANE | 5000 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.48 Data Food pada Database MySQL sebelum update data
Sumber : *Pengujian Dan Analisis*

Hasil update data food berupa harga makanan yang telah dimasukkan pada database MySQL melalui Sistem Transaksi Pembayaran diperlihatkan dalam Gambar 6.49.

```
mysql> select food_id, food_name, price from tb_food
-> where food_name='Roti Cane';
+-----+-----+-----+
| food_id | food_name | price |
+-----+-----+-----+
| s01-04 | ROTI CANE | 6500 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 6.49 Hasil Pengujian Edit Data Food pada Database MySQL
Sumber : *Pengujian Dan Analisis*

Berdasarkan tampilan diatas, tampak bahwa perubahan data harga makanan berhasil disimpan dengan baik pada tb_food dalam database MySQL.

6.4.3 Pengujian Proses Report Data Food

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah Sistem Transaksi Pembayaran FoodCourt dapat melakukan proses menampilkan report data food yang tersimpan didalam database foodcourt.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** |.
- Memilih menu |**Report**| pada Food Master Data.
- Menampilkan data food pada database MySQL dengan perintah:
SELECT food_id, food_name, price FROM tb_food
WHERE (delete_flag ='N' && counter_id='s01');
- Menampilkan data food pada database MySQL dengan perintah:
SELECT food_id, food_name, price FROM tb_food
WHERE (delete_flag ='Y' && counter_id='s01');

c. Hasil Pengujian dan Analisis

Antarmuka Report pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.50.

COUNTER ID	COUNTER NAME	FOOD ID	FOOD NAME	PRICE	DEL?
s01	Kebab Ali Baba				
		s01-04	ROTI CANE	6.500	N
		s01-03	TEH TARIK	4.000	N
		s01-02	KEBAB SEDANG	10.000	N
		s01-01	KEBAB SUPER	15.000	N

Gambar 6.50 Antarmuka Report Food
 Sumber : Pengujian Dan Analisis

Report data food diperoleh berdasarkan delete flag. Apabila status makanan masih disediakan oleh counter makanan maka delete flag-nya adalah N (No) begitu pula sebaliknya, status makanan yang sudah tidak tersedia pada counter makanan yang bersangkutan maka delete flag-nya adalah Y (Yes). Berikut hasil query data makanan yang tersedia yang tersimpan pada tabel tb_food dalam database foodcourt diperlihatkan pada Gambar 6.51.

```
mysql> SELECT food_id, food_name, price FROM tb_food
-> WHERE (delete_flag = 'N' && counter_id='s01');
```

food_id	food_name	price
s01-04	ROTI CANE	6500
s01-03	TEH TARIK	4000
s01-02	KEBAB SEDANG	10000
s01-01	KEBAB SUPER	15000

4 rows in set (0.00 sec)

Gambar 6.51 Hasil Pengujian Report Food dengan Delete Flag "N/No"
 Sumber : Pengujian Dan Analisis

Sedangkan hasil query data makanan yang sudah tidak tersedia yang tersimpan pada tabel tb_food dalam database foodcourt diperlihatkan pada Gambar 6.52.

```
mysql> SELECT food_id, food_name, price FROM tb_food  
-> WHERE (delete_flag = 'Y' && counter_id='s01');  
Empty set (0.00 sec)
```

Gambar 6.52 Hasil Pengujian Report Food dengan Delete Flag "Y/Yes"
Sumber : *Pengujian Dan Analisis*

Berdasarkan tampilan diatas, dapat diketahui bahwa pada tb_food dalam database foodcourt tidak terdapat data makanan yang sudah tidak tersedia.

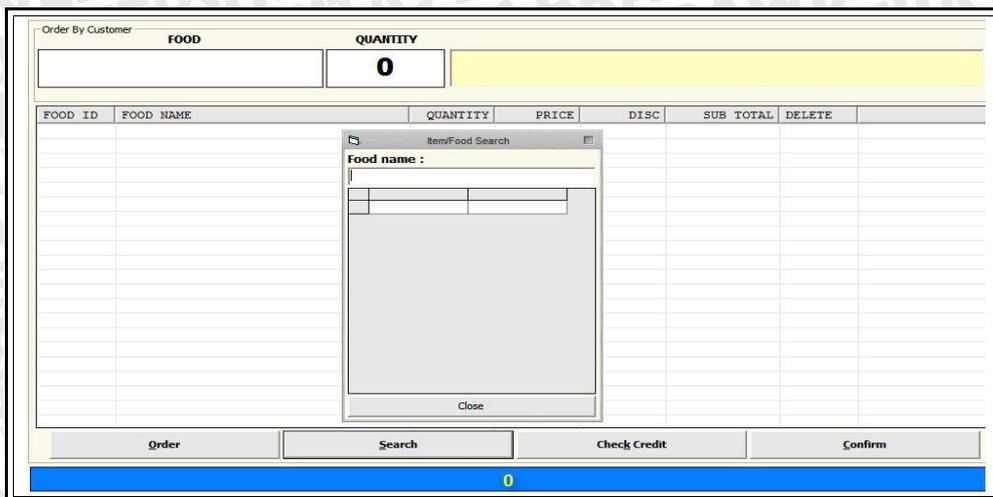
6.4.4 Pengujian Proses Order Makanan

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah Sistem Transaksi Pembayaran FoodCourt dapat melakukan proses order makanan dan menyimpan rekapan transaksi kedalam database foodcourt.

b. Prosedur Pengujian

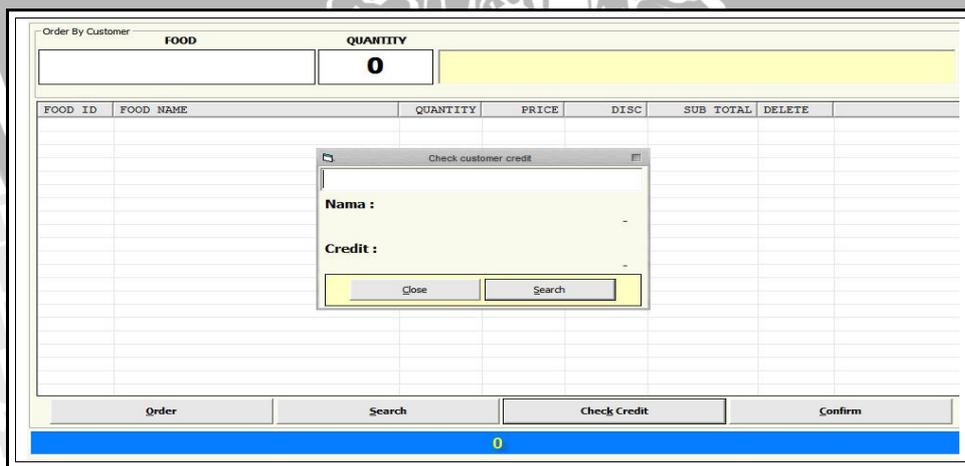
- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada | **Path Aplikasi|FoodCourtSys.exe** |.
- Memilih menu |**Order By Customer**|.
- Memilih menu |**Search**| pada halaman order by customer, maka akan tampil form seperti ditunjukkan dalam Gambar 6.53.



Gambar 6.53 Antarmuka Food Search

Sumber : Pengujian Dan Analisis

- Masukkan nama makanan pada form search, misalny: “Kebab”.
- Memilih nama makanan yang ditampilkan dalam list view dengan mengklik ganda pada baris tersebut.
- Masukkan jumlah makanan yang dipesan pada textbox “Quantity”
- Memilih menu |**Order**| pada halaman order by customer.
- Memilih menu |**Check Credit**| pada halaman order by customer, maka akan tampil form seperti yang diperlihatkan dalam Gambar 6.54.



Gambar 6.54 Antarmuka Check Credit

Sumber : Pengujian Dan Analisis

- Memilih menu |**Confirm**| pada halaman order by customer.

- Menampilkan daftar transaksi yang telah dikonfirmasi dengan memberi perintah :

```
SELECT trans_id, food_id, qty, price
FROM tb_trans_dtl;
```

c. Hasil Pengujian dan Analisis

Antarmuka Order By Customer pada Sistem Transaksi Pembayaran Food Court ditunjukkan dalam Gambar 6.55.

FOOD ID	FOOD NAME	QUANTITY	PRICE	DISC	SUB TOTAL	DELETE
		0				

Gambar 6.55 Antarmuka Order By Customer
Sumber : *Pengujian Dan Analisis*

- Pada menu *search* dengan memberikan masukan berupa nama makanan (apabila nama makanan sesuai dengan data yang tersimpan dalam database) maka akan ditampilkan nama makanan dan ID makanan seperti yang diperlihatkan pada Gambar 6.56.

Order By Customer

FOOD	QUANTITY
	0

FOOD ID	FOOD NAME	QUANTITY	PRICE	DISC	SUB TOTAL	DELETE

ItemFood Search

Food name : Kebab |

food_id	food_name
s01-02	KEBAB SEDANG
s01-01	KEBAB SUPER

Close

Order Search Check Credit Confirm

0

Gambar 6.56 Form Search pada halaman Order by Customer

Sumber : Pengujian Dan Analisis

- Pada form *food/item search*, ketika menu makanan dipilih dengan mengklik ganda pada baris tersebut, maka akan ditampilkan ID dan nama makanan, kemudian quantity diisi berdasarkan banyaknya jumlah makanan yang dipesan seperti diperlihatkan dalam Gambar 6.57.

Order By Customer

FOOD	QUANTITY
s01-01	2

FOOD ID	FOOD NAME	QUANTITY	PRICE	DISC	SUB TOTAL	DELETE

Order Search Check Credit Confirm

Gambar 6.57 halaman Order by Customer

Sumber : Pengujian Dan Analisis

- Pada proses *order*, ID dan nama makanan serta quantities akan tersimpan dalam list order seperti diperlihatkan dalam Gambar 6.58

Order By Customer

FOOD	QUANTITY
	0

FOOD ID	FOOD NAME	QUANTITY	PRICE	DISC	SUB TOTAL	DELETE
s01-01	KEBAB SUPER	2	15000	0	30000	X

Order Search Check Credit Confirm

30000

Gambar 6.58 Rekapitan order pada list order
Sumber : *Pengujian Dan Analisis*

Apabila *quantity* tidak terisi atau bernilai 0, maka sistem tidak dapat menyimpan order kedalam list order.

- Pada proses *Check Credit*, masukan berupa ID customer yang diperoleh dari proses *scanning E-card* akan menampilkan nama customer serta sisa kredit yang dimiliki customer. Form *Check Credit* pada halaman *Order by Customer* diperlihatkan dalam Gambar 6.59.

FOOD ID	FOOD NAME	QUANTITY	PRICE	DISC	SUB TOTAL	DELETE
#01-01	KEBAB SUPER	2	15000	0	30000	NO

Check customer credit

0809140002

Nama : Nesya

Credit : 168.000

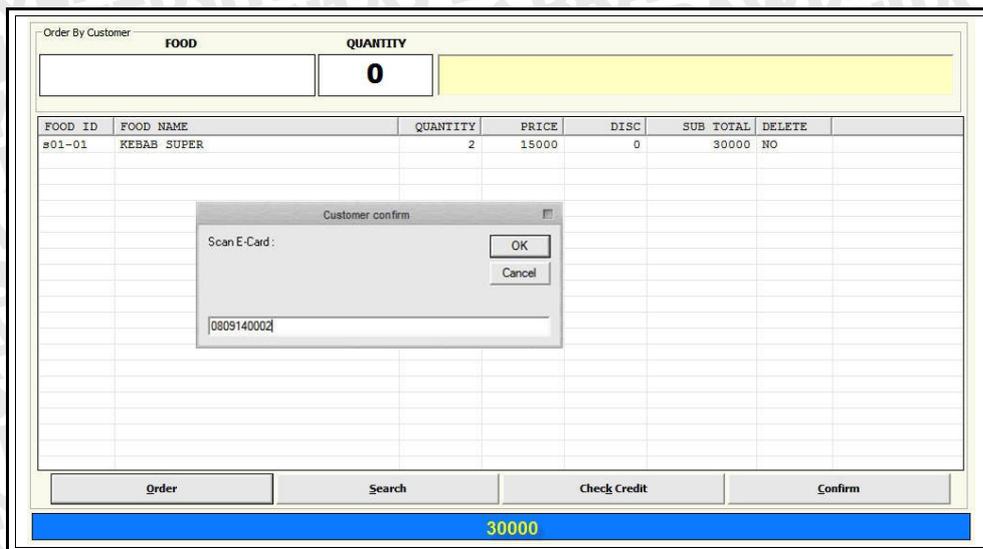
Close Search

30000

Gambar 6.59 Form *Check Credit* pada halaman *Order by Customer*
Sumber : *Pengujian Dan Analisis*

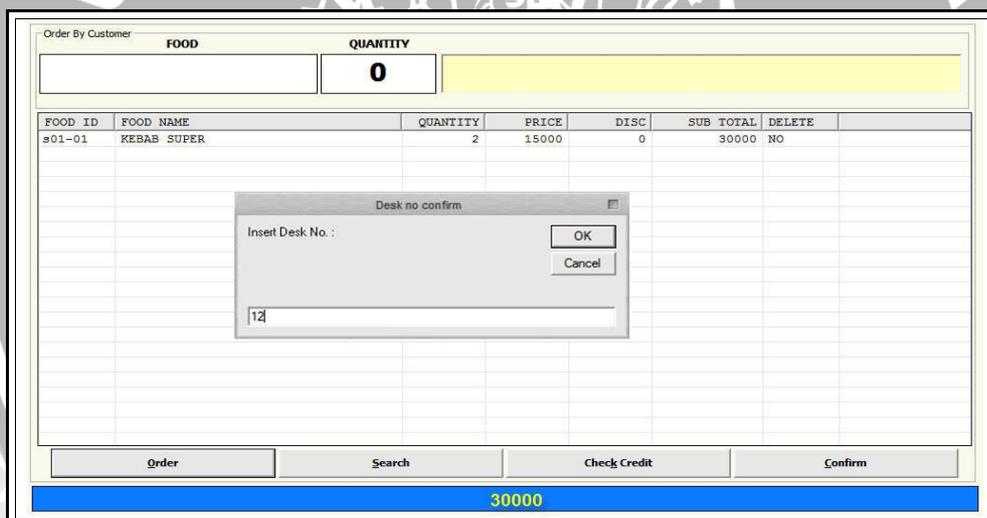
Apabila sisa kredit yang dimiliki customer mencukupi, maka transaksi pembayaran dapat dilakukan. Jika sisa kredit tidak memenuhi, maka customer harus melakukan isi ulang kredit.

- Pada proses *confirm*, dengan pilihan *Yes*, akan ditampilkan form yang akan memuat ID customer yang diperoleh dari proses *scanning E-card* dilanjutkan dengan menentukan nomor meja yang ditempati oleh customer. Proses *scan E-card* pada menu *confirm* diperlihatkan dalam Gambar 6.60.



Gambar 6.60 Scan E-card pada Menu Confirm
Sumber : *Pengujian Dan Analisis*

Sedangkan proses konfirmasi nomor meja diperlihatkan dalam Gambar 6.61.



Gambar 6.61 Konfirmasi nomor meja pada Menu Confirm
Sumber : *Pengujian Dan Analisis*

Setelah konfirmasi selesai, maka sistem akan menyimpan transaksi ke dalam *database*.

- Hasil query dari daftar order makanan yang telah dikonfirmasi dan tersimpan dalam *database* diperlihatkan dalam Gambar 6.62

```
mysql> select trans_id, food_id, qty, price
-> from tb_trans_dtl;
```

trans_id	food_id	qty	price
1	s01-04	1	5000
1	s01-01	1	15000
1	s01-03	1	4000
2	s02-02	2	8000
2	s02-04	2	3000
2	s02-05	2	5000
3	s02-03	1	8000
3	s02-01	1	10000
3	s02-04	2	3000
4	s03-02	1	25000
4	s03-01	1	25000
4	s03-06	2	5000
4	s03-03	1	10000
5	s01-01	2	15000

14 rows in set (0.00 sec)

Gambar 6.62 Hasil Pengujian Daftar transaksi
Sumber : Pengujian Dan Analisis

Transaksi dengan ID = '5' merupakan transaksi yang dilakukan dalam pengujian, tampak bahwa data yang terdapat pada gambar tersebut sesuai dengan data yang dimasukkan pada proses order makanan.

6.4.5 Pengujian Proses Delivery Check

a. Tujuan

Pengujian ini dilakukan untuk melihat apakah Sistem Transaksi Pembayaran Food Court dapat melakukan proses menampilkan data pesanan atau order customer dan melakukan perubahan status order customer ketika order telah terpenuhi.

b. Prosedur Pengujian

- Menjalankan aplikasi Sistem Transaksi Pembayaran Food Court pada |**Path Aplikasi|FoodCourtSys.exe**|.
- Memilih menu |**Delivery Check**|.
- Mengubah status order customer dengan mengklik ganda pada baris yang diinginkan.
- Menampilkan status order customer sebelum diubah dengan memberi perintah:

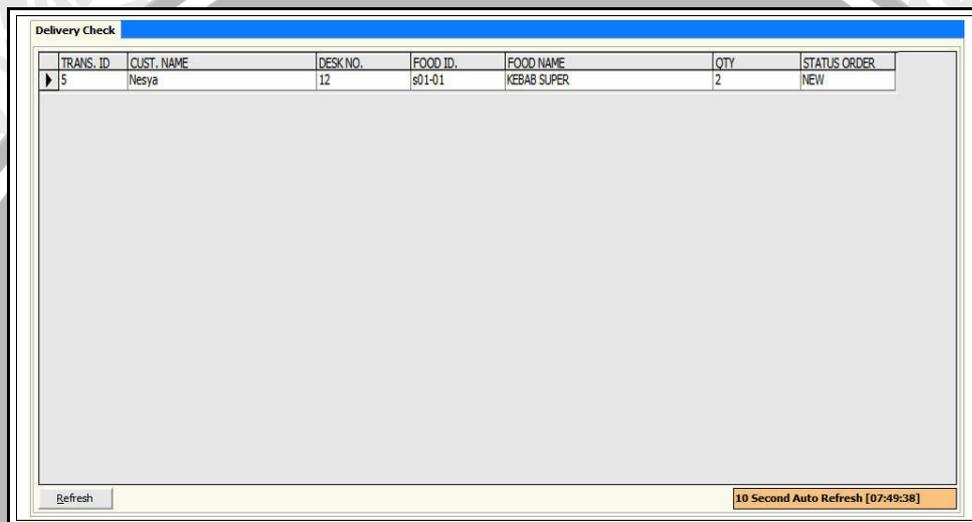
```
SELECT trans_id, food_id, qty, status_order FROM
tb_trans_dtl;
```

- Menampilkan status order customer setelah diubah dengan memberi perintah:

```
SELECT trans_id, food_id, qty, status_order FROM  
tb_trans_dtl;
```

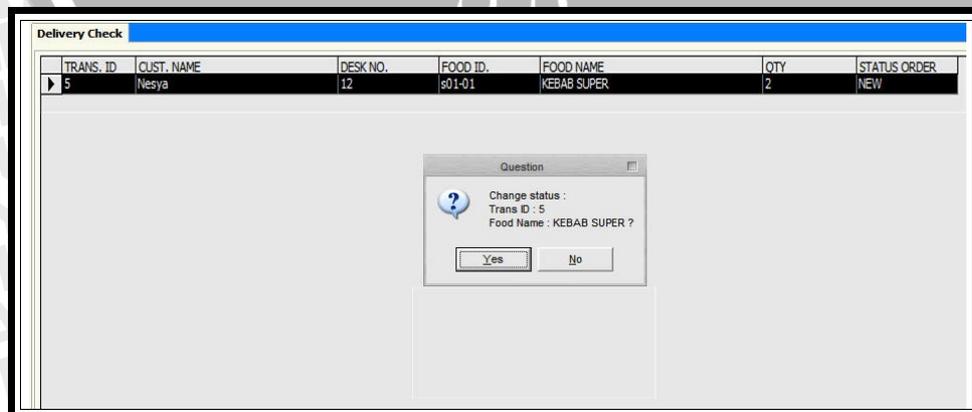
c. Hasil Pengujian dan Analisis

Antarmuka Delivery Check pada Sistem Transaksi Pembayaran Food Court diperlihatkan dalam Gambar 6.63.



Gambar 6.63 Antarmuka Delivery Check
Sumber : *Pengujian Dan Analisis*

Pada proses perubahan status order yaitu dengan mengklik ganda pada record order akan tampil pilihan untuk perubahan status seperti di perlihatkan dalam Gambar 6.64.



Gambar 6.64 Perubahan status order

Sumber : *Pengujian Dan Analisis*

Sistem akan secara otomatis menghapus order yang telah mengalami perubahan status, karena sistem diset untuk mereshfresh table setiap 10 detik, sehingga tidak terjadi penumpukan data yang dapat menghambat jalannya sistem.

Hasil query status order sebelum dilakukan perubahan status diperlihatkan dalam Gambar 6.65.

```
mysql> select trans_id, food_id, qty, status_order
-> from tb_trans_dtl
-> ;
```

trans_id	food_id	qty	status_order
1	s01-04	1	DELIVER
1	s01-01	1	DELIVER
1	s01-03	1	DELIVER
2	s02-02	2	DELIVER
2	s02-04	2	DELIVER
2	s02-05	2	DELIVER
3	s02-03	1	DELIVER
3	s02-01	1	DELIVER
3	s02-04	2	DELIVER
4	s03-02	1	DELIVER
4	s03-01	1	DELIVER
4	s03-06	2	DELIVER
4	s03-03	1	DELIVER
5	s01-01	2	NEW

14 rows in set (0.00 sec)

Gambar 6.65 Hasil pengujian sebelum perubahan status order

Sumber : *Pengujian Dan Analisis*

Hasil query status order setelah dilakukan perubahan status diperlihatkan dalam Gambar 6.66

```
mysql> select trans_id, food_id, qty, status_order
-> from tb_trans_dtl;
```

trans_id	food_id	qty	status_order
1	s01-04	1	DELIVER
1	s01-01	1	DELIVER
1	s01-03	1	DELIVER
2	s02-02	2	DELIVER
2	s02-04	2	DELIVER
2	s02-05	2	DELIVER
3	s02-03	1	DELIVER
3	s02-01	1	DELIVER
3	s02-04	2	DELIVER
4	s03-02	1	DELIVER
4	s03-01	1	DELIVER
4	s03-06	2	DELIVER
4	s03-03	1	DELIVER
5	s01-01	2	DELIVER

14 rows in set (0.00 sec)

Gambar 6.66 Hasil pengujian setelah perubahan status order

Sumber : *Pengujian Dan Analisis*

1.1. Pengujian Pembacaan barcode pada E-card



a. Tujuan

Pengujian ini dilakukan untuk mengetahui kondisi ideal (berdasarkan jarak dan sudut kemiringan *barcode reader* terhadap sample *barcode*) dalam proses pembacaan / *scanning barcode*.

b. Prosedur Pengujian

- Menjalankan program Notepad pada | **Start** | **Program** | **Accessories** | **Notepad** |.
- *Barcode reader* ditembakkan pada satu sample *barcode* sebanyak 20 kali dengan jarak dan sudut kemiringan tertentu. Data yang akan digunakan pada pengujian pembacaan *barcode* dapat dilihat pada Tabel 6.6.

Tabel 6.6 Data Pengujian Pembacaan Barcode

Sample barcode	Jarak (dalam cm)	Sudut Kemiringan (dalam θ°)
0809010004	0	30
	1	45
0809010002	2	60
	3	90

Sumber : *Pengujian Dan Analisis*

c. Hasil Pengujian dan Analisis

Data hasil pengujian pembacaan barcode berdasarkan jarak dan sudut kemiringan tertentu ditunjukkan pada Tabel 6.7.

Tabel 6.7 Data Hasil Pengujian Pembacaan Barcode

		Sudut Kemiringan (dalam θ°)			
		30	45	60	90
Jarak (dalam cm)	0	✓	✓	✓	✓
	1	✓	✓	✓	✓
	2	✓	✓	✓	✓
	3	✓	✓	✗	✗

Sumber : *Pengujian Dan Analisis*

Keterangan :

- ✓ : barcode dapat terbaca dengan baik
- ✗ : barcode tidak dapat terbaca



Data hasil pengujian pembacaan barcode yang dicantumkan pada Tabel 6.7 merupakan jarak dan sudut kemiringan ideal yang dibutuhkan oleh barcode reader untuk dapat membaca barcode yang terdapat pada E-card dengan baik yaitu dengan jarak antara 0-2 cm dengan sudut kemiringan antara 30° - 90° .

