

# Sistem Transaksi Pembayaran Food Court Dengan Teknologi E-Card

Heru Nurwasito

*Food court center* merupakan salah satu fasilitas yang sering kita jumpai pada pusat-pusat perbelanjaan, akan tetapi masih banyak *food court* yang memiliki sistem pembayaran yang kurang baik. Dalam upaya mengatasi permasalahan tersebut, maka pada penelitian ini dirancang dan dibuat perangkat lunak Sistem Transaksi Pembayaran *Food Court* dengan menggunakan teknologi *E-card* yang dapat menyediakan kemudahan dalam transaksi pembayaran dan manajemen pengelolaan *food court*. Pengembangan Sistem Transaksi Pembayaran *Food Court* dilakukan dengan menggunakan *development tool* Visual Basic 6.0 dan basis data MySQL. Sistem Transaksi Pembayaran *Food Court* dibangun dengan menggunakan arsitektur *Client-Server* sehingga keakuratan data dapat terjaga karena semua informasi disimpan dalam satu komputer. Pengujian Sistem Transaksi Pembayaran *Food Court* dilakukan pada setiap aplikasi sistem untuk mengetahui proses yang dilakukan oleh tiap-tiap aplikasi sistem tersebut. Hasil dari pengujian aplikasi sistem dapat diketahui bahwa aplikasi sistem pada Sistem Transaksi Pembayaran *Food Court* dapat melakukan proses sesuai dengan kegunaannya masing-masing. Pengujian juga dilakukan terhadap koneksi basis data dan pembacaan *barcode* pada *E-card*. Sistem Transaksi Pembayaran *Food Court* yang dihubungkan dengan koneksi basis data MySQL pada *port* 3306 dapat melakukan proses manipulasi terhadap data-data di dalam basis data. Pengujian pembacaan *barcode* dilakukan dengan menembakkan *barcode reader* pada *sample barcode* sebanyak 20 kali dengan jarak dan sudut kemiringan tertentu. Hasil pengujian yang dilakukan terhadap koneksi basis data dan pembacaan *barcode* menunjukkan bahwa Sistem Transaksi Pembayaran *Food Court* dapat berfungsi dengan baik.

**Kata Kunci :** transaksi pembayaran, *food court*, *E-card*, *barcode*, *barcode reader*, Visual Basic, MySQL.

## I. PENDAHULUAN

Perekonomian di Indonesia berkembang sangat pesat, terbukti dari maraknya pembangunan mall dikota-kota besar, hal tersebut tentu saja mempengaruhi gaya hidup masyarakat perkotaan yang cenderung konsumtif. Oleh karena itu, para pengembang berlomba-lomba untuk memberikan fasilitas-fasilitas yang terbaik demi kenyamanan pengunjung.

*Food court center* merupakan salah satu fasilitas yang disediakan oleh pihak pengembang, akan tetapi sering kita jumpai antrian yang terjadi pada loket

kasir, hal tersebut tentu saja dapat mengurangi kenyamanan pelanggan dan mengakibatkan kelalaian petugas kasir. Terdapat pula beberapa *food court* yang sistem pembayarannya tidak terpusat pada loket kasir tetapi melakukan pembayaran langsung pada *counter* makanan. Hal tersebut dapat menimbulkan masalah apabila pihak *counter* makanan melakukan kecurangan dengan memanipulasi laporan keuangan, sehingga pada saat pembagian pendapatan, pihak pengelola tidak mendapatkan apa yang menjadi haknya.

Untuk mengatasi masalah tersebut maka dibuatlah sistem transaksi pembayaran *food court* yang dapat mendukung kenyamanan pelanggan pada saat melakukan transaksi pembayaran dan juga memberikan kemudahan dalam manajemen pengelolaan *food court*. Sistem ini akan menerapkan penggunaan media kartu elektronik, atau disebut *E-card* yang berbasis *barcode*. Karena pada sistem ini, *E-card* akan menggantikan uang tunai sebagai alat bantu pembayaran, sehingga kita tidak perlu membawa uang tunai dalam jumlah yang cukup banyak sekaligus dapat mempercepat transaksi pembayaran karena pelanggan tidak perlu mengantri di loket kasir. Sistem Transaksi Pembayaran *Food Court* akan dibangun dengan menggunakan Visual Basic dan basisdata MySQL.

## II. TRANSAKSI KEUANGAN

Transaksi keuangan merupakan kegiatan ekonomi pada suatu unit organisasi atau kejadian yang menyangkut unit organisasi yang harus diukur jumlah rupiahnya dan dicatat dalam sistem akuntansi sehingga jumlah rupiahnya akan mempengaruhi laporan keuangan yang dihasilkan [SUW – 91]. Untuk dapat dikatakan sebagai transaksi keuangan, transaksi tersebut harus dapat dinyatakan dalam satuan uang. Untuk dapat disebut sebagai transaksi, kegiatan tersebut tidak harus berupa kegiatan fisik. Hasil suatu keputusan manajemen yang berpengaruh terhadap laporan keuangan juga termasuk dalam pengertian transaksi.

Untuk suatu unit organisasi berupa perusahaan, Transaksi keuangan dapat digolongkan menjadi transaksi Eksternal dan transaksi Internal. Transaksi Eksternal adalah transaksi yang terjadi antara perusahaan dengan pihak luar sedangkan Transaksi Internal adalah transaksi yang terjadi dalam unit perusahaan.

Pemrosesan data menyangkut berbagai tindakan dan kegiatan, antara lain :

Heru Nurwasito adalah dosen Jurusan Teknik Elektro Univerrritas Brawijaya dengan alamat korespondensi Jurusan Teknik Elektro Universitas Brawijaya, Jl. MT. Haryono 167 Malang telp. 0341-665144

- *Mengidentifikasi* berarti menentukan apakah suatu objek dalam transaksi memenuhi kriteria untuk diukur dan dimasukkan dalam laporan keuangan.
- *Mengukur* berarti menentukan jumlah rupiahnya agar jumlah rupiah tersebut dapat diolah lebih lanjut.
- *Mencatat* berarti memasukkan angka hasil pengukuran kedalam sistem akuntansi sehingga angka kos tersebut termuat dalam laporan keuangan.
- *Mengklasifikasi* berarti memasukkan suatu jumlah rupiah kedalam kategori yang telah ditentukan.
- *Mengalokasi* berarti memecah kesatuan kos menjadi beberapa bagian berdasarkan objek tertentu.
- *Meringkas* berarti menggabungkan berbagai jumlah rupiah dari berbagai kategori khusus kedalam kategori yang lebih umum.
- *Menyajikan* berarti memuat jumlah rupiah hasil penggolongan kedalam laporan dengan format dan susunan yang telah ditentukan.

### III. BARCODE READER

Komputer tidak dapat membaca *barcode* secara langsung. Untuk mengetahui informasi yang terdapat pada *barcode*, *barcode* harus didekodekan kedalam format data yang dapat diproses oleh komputer. Alat yang dapat membaca informasi yang terdapat pada *barcode* disebut *barcode reader*, atau secara umum disebut *barcode scanner*.

*Barcode reader* terdiri atas :

- *Scanner* berfungsi untuk men-*scan* symbol *barcode* dan menangkap bar dan spasi dari *barcode* dan mengirimnya ke *decoder*.
- *Decoder* berfungsi sebagai penerjemah bar dan spasi kedalam bentuk karakter ASCII dan mentransmisikan data ke komputer dalam format data sederhana.
- Kabel berfungsi sebagai antarmuka antara dekoder dan komputer.

Beberapa tipe *Barcode Reader* antara lain adalah:

- *Barcode reader* berbentuk pena atau tongkat merupakan *barcode reader* yang paling sederhana dan tidak begitu mahal. Paling tahan lama dan tidak terdiri atas bagian-bagian yang bergerak..
- *Laser barcode scanners* merupakan *barcode scanner* paling populer. *Laser barcode scanner* tidak perlu menyentuh atau mendekati *barcode* untuk pembacaan *barcode*.
- CCD (*Charge Couple Devices*) *barcode scanner* menggunakan array cahaya sensor yang terdapat pada bagian depan *barcode reader*.
- Kamera berbasis *barcode reader* menggunakan video kamera kecil untuk menangkap gambar *barcode*.

## IV. PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan perancangan Sistem Transaksi Pembayaran *Food Court* dengan media *E-card* yang berbasis *barcode* dengan menggunakan kompuler Microsoft Visual Basic 6.0 dengan bahasa pemrograman basic dan database MySQL. Perancangan yang dilakukan meliputi dua tahap, yaitu : analisis kebutuhan dan perancangan perangkat lunak (desain).

### A. Spesifikasi Kebutuhan

Spesifikasi kebutuhan diperlukan untuk menjelaskan kebutuhan perangkat lunak yang telah didefinisikan sebelumnya secara lebih detail dan tepat yang akan menjadi dasar bagi perancangan dan implementasi. Spesifikasi kebutuhan dari Sistem Transaksi Pembayaran *Food Court* ditunjukkan dalam Tabel 1.

TABEL 1. SPESIFIKASI KEBUTUHAN

Kebutuhan	Use Case
Untuk memasukkan data pribadi <i>customer</i> , <i>password</i> serta pengaktifasian <i>E-card</i>	Registrasi
Untuk mengetahui sisa kredit yang terdapat pada <i>E-card</i>	Cek kredit
Untuk mengisi ulang kredit pada <i>E-card</i>	Refill kredit
Untuk melakukan pemesanan makanan pada <i>counter</i> makanan	Order makanan
Untuk melakukan pembayaran pada <i>counter</i> makanan	Pembayaran
Untuk mengetahui transaksi yang telah dilakukan oleh <i>customer</i>	Laporan transaksi
Untuk meng- <i>update</i> menu makanan	Update menu
Untuk mengubah status order <i>customer</i>	Ubah status
Untuk mengetahui rekapitulasi hasil penjualan	Laporan keuangan
Untuk mengetahui setiap transaksi yang telah terjadi pada <i>Food Court</i>	Monitoring transaksi

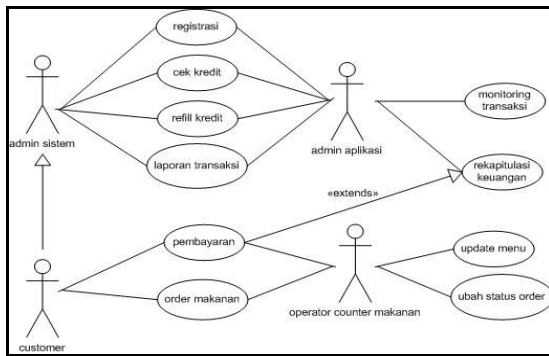
### B. Use Case Diagram

Dari rumusan kebutuhan – kebutuhan di atas, dimodelkan dalam *use case diagram*. Pada Tabel 2 dibawah ini memberikan uraian pada masing-masing aktor yang digunakan dalam diagram use case.

TABEL 2. DESKRIPSI ACTOR

Actor	Deskripsi Actor
Customer	Pengguna yang akan menggunakan sistem ini untuk melakukan order makanan dan pembayaran langsung pada counter makanan.
Admin Sistem	Pengguna yang akan menggunakan sistem ini untuk membantu customer melakukan registrasi, pengecekan kredit, dan refill kredit, serta laporan transaksi yang telah dilakukan oleh customer
Operator Counter makanan	Pengguna yang akan menggunakan sistem ini untuk melakukan update menu, melayani order, dan menerima pembayaran.
Admin Aplikasi	Pengguna yang akan menggunakan system ini untuk melakukan monitoring transaksi, dan rekapitulasi transaksi

Pada gambar 1 dibawah ini diperlihatkan pemodelan dalam *diagram use case* yang menggambarkan interaksi yang dilakukan *customer*, admin sistem, admin aplikasi dan operator *counter* terhadap sistem.

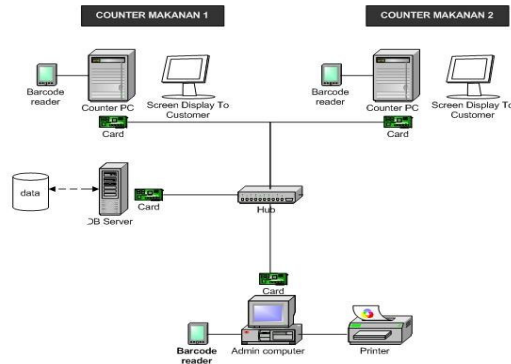


Gambar 1 Use Case Diagram Sistem Transaksi Pembayaran Food Court

C. Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang akan dibuat secara umum. Perancangan sistem meliputi diagram blok sistem, cara kerja sistem, dan diagram konteks.

Diagram blok sistem menggambarkan setiap blok atau bagian dari sistem aplikasi. Sistem Transaksi Pembayaran *Food Court* dirancang untuk dapat dijalankan pada jaringan lokal. Sistem aplikasi akan dibagi menjadi beberapa bagian yang terhubung dengan *server*, seperti yang diperlihatkan dalam Gambar 2 :



Gambar 2 Diagram blok sistem

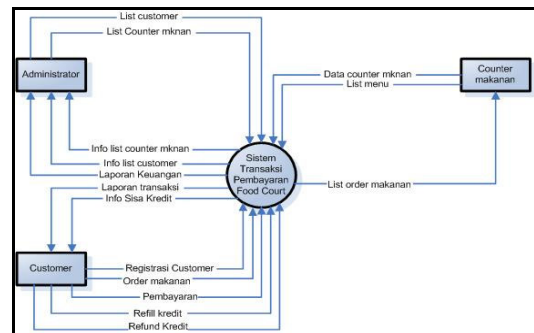
Sistem Transaksi Pembayaran *Food Court* terdiri dari *hardware* dan *software* yang saling mendukung. Hardware yang digunakan yaitu *barcode reader*, berfungsi sebagai alat untuk menkodekan *barcode* yang terdapat pada *E-card*. Prinsip kerja dari sistem ini diawali dari proses registrasi *customer* yang belum memiliki *E-card*. Pada saat proses registrasi, *E-card* akan diaktivasi dengan penggunaan *password* oleh *customer*. Setelah pengaktifasian, *customer* dapat menggunakan *E-card* pada *counter-counter* makanan yang tersedia pada *food court* yang menggunakan sistem tersebut.

Bagi *customer* yang telah memiliki *E-card* atau sudah melakukan registrasi, maka dapat dilakukan pengecekan sisa kredit dan melakukan pengisian ulang kredit tersebut. Pada saat *customer* melakukan order makanan, transaksi pembayaran langsung dilakukan pada *counter* makanan tersebut.

Operator *counter* akan membaca *barcode* pada *E-card* melalui *barcode reader* dan melihat sisa kredit serta transaksi yang telah dilakukan. Pada saat *customer* menunggu pesanan, Operator *counter* dapat melihat status order pelanggan yang selanjutnya akan menginformasikan kepada pelayan. Jika diinginkan, *customer* dapat mencairkan sisa kredit yang ada pada *E-card* tersebut setelah *customer* melakukan transaksi pembayaran.

DFD yang pertama kali dibuat adalah diagram konteks (*context diagram*). Diagram konteks menggambarkan seluruh input ke sistem atau output dari sistem serta menjelaskan hubungan sistem dengan lingkungan atau kesatuan luar. Pada sistem ini, melibatkan 3 kesatuan luar, yaitu administrator, *customer*, dan *counter* makanan.

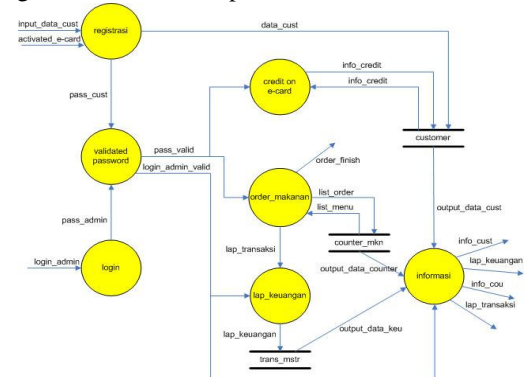
Diagram konteks ditunjukkan dalam Gambar 3.



Gambar 3. Diagram Konteks Sistem

*Data Flow Diagram* (DFD) merupakan salah satu alat yang digunakan untuk menggambarkan keseluruhan Sistem Transaksi Pembayaran *Food Court* yang akan dibuat. DFD yang pertama kali dibuat adalah DFD level 0 atau *Context Diagram*. DFD *context diagram* telah dijelaskan sebelumnya pada perancangan sistem dengan bahasan diagram konteks.

Tahap selanjutnya yang merupakan penjabaran diagram konteks adalah pembuatan DFD level 0.



Gambar 4. DFD Level 0

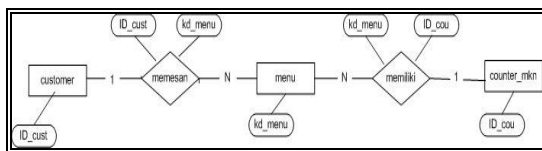
Gambar 4. merupakan DFD level 0 dari Sistem Transaksi Pembayaran *Food Court* yang mempunyai tujuh proses, yaitu : registrasi, validasi password, login, kredit pada *E-card*, order makanan, laporan keuangan, dan informasi.

Perancangan *database* dilakukan agar Sistem Transaksi Pembayaran Food Court menjadi *database* yang efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam pemanipulasian data. Perancangan *database* dapat dilakukan dengan menggunakan *Entity Relationship Diagram* (Diagram ER), normalisasi data, dan *Data Object Descripton*

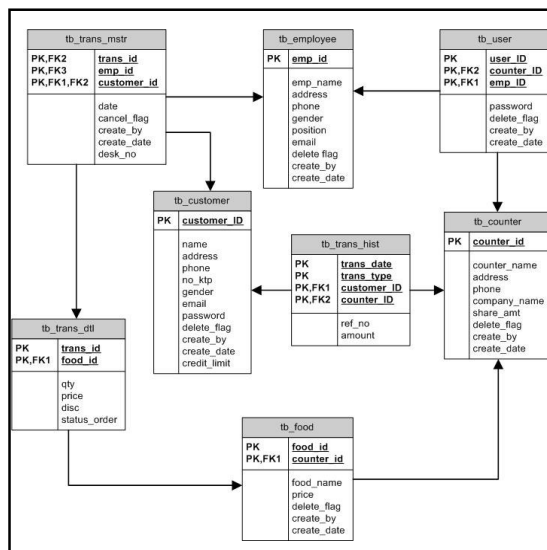
Diagram ER digunakan untuk menggambarkan entitas-entitas secara umum yang terdapat pada Sistem Transaksi Pembayaran *Food Court*. Akan tetapi perancangan database dengan diagram ER tidak sepenuhnya mewakili struktur yang terdapat pada keadaan nyata dari Sistem Transaksi Pembayaran *Food Court*.

Diagram ER akan menggambarkan hubungan entitas satu dengan lainnya dengan memperlihatkan hubungan antar atribut yang akan dijadikan *key* untuk berelasi antar tabel. Sebelum menggambarkan relasi diagram ER, entitas-entitas pembentuk sistem harus ditentukan terlebih dahulu beserta atribut entitas itu sendiri. Entitas yang ditentukan merupakan tabel yang akan dipakai pada database.

Diagram ER dari struktur *database* foodcourt digambarkan dalam Gambar 5 berikut:



Gambar 5. Entity Relationship Diagram



Gambar 6. Normalisasi pada database foodcourt

Pada Sistem Transaksi Pembayaran Food Court, normalisasi data dibutuhkan untuk mengurangi pengulangan (redundansi) data di setiap tabel dan mempercepat proses pencarian data di dalam *database*. Tabel-tabel yang akan digunakan untuk menyimpan data dinormalisasi menjadi enam buah entitas (tabel) yaitu Tabel *tb\_customer*, *tb\_counter*, *tb\_food*, *tb\_employee*,

*tb\_trans\_mstr*, *tb\_trans\_dtl*, *tb\_trans\_hist*, dan *tb\_user*. Rancangan normalisasi tabel-tabel tersebut diperlihatkan dalam Gambar 6.

Implementasi perancangan basis data foodcourt dilakukan sesuai dengan *entity relationship diagram*. Implementasi perancangan basis data foodcourt menggunakan *Data Definition Language* (DDL).

DDL yang digunakan dalam membentuk basis data foodcourt adalah sebagai berikut:

```
CREATE DATABASE foodcourt ;
```

Basis data foodcourt yang telah diimplementasikan pada basis data MySQL ditunjukkan dalam Gambar 7.

Gambar 7. Implementasi Basis Data foodcourt pada MySQL

Pada implementasi perangkat lunak aplikasi dilakukan implementasi perangkat lunak untuk *login*, master data employee, master data customer, master data counter makanan, master data food/makanan, order customer, delivery check, report dan user management.

## V. PENGUJIAN

Pengujian yang dilakukan meliputi pengujian koneksi basisdata, pengujian implementasi Sistem Transaksi Pembayaran Food Court, dan pengujian pembacaan barcode pada E-card.

### A. Pengujian Koneksi Basis Data

Gambar 8. 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 8. Daftar Koneksi Pada Komputer Client Sebelum Melakukan Koneksi dengan Database MySQL

Gambar 8. menunjukkan hasil dari penggunaan perintah 'netstat -an', perintah tersebut digunakan untuk menampilkan semua koneksi yang terdapat pada komputer *client*. 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 9.

```
C:\Documents and Settings\nasisten>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    0.0.0.0:1070            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    192.10.10.1:139         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    0.0.0.0:500             *:*
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    192.10.10.1:18708      *:*
UDP    192.10.10.1:59163      *:*
```

Gambar 9. Daftar Koneksi Pada Komputer Client Setelah Melakukan Koneksi dengan Database MySQL

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.

**B. Pengujian Implementasi Perangkat Lunak Sistem Transaksi Pembayaran Food Court**

Hasil dari pengujian ini akan menampilkan data yang terdapat pada *database* foodcourt.

**1) Tabel tb\_customer**

Tabel *tb\_customer* digunakan untuk menyimpan data customer yang telah terdaftar.

Untuk menampilkan data customer digunakan perintah:

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

Hasil perintah tersebut ditunjukkan dalam Gambar 10.

```
mysql> select cust_id, cust_name, address, credit_limit, phone from tb_customer;
+-----+-----+-----+-----+-----+
| cust_id | cust_name | address | credit_limit | phone |
+-----+-----+-----+-----+-----+
| 0009140001 | Vento | Jl. Cempaka 33 | 100000 | 0341458761 |
| 0009140002 | Nerya | Jl. Kawajaya 53 | 150000 | 0823490595 |
| 0009140003 | Redityo | Jl. Teratai 196 | 100000 | 058218610600 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Gambar 10. Data Tabel *tb\_customer*

**2) Tabel tb\_employee**

Tabel *tb\_employee* digunakan untuk menyimpan data employee yang bekerja pada foodcourt tersebut.

Untuk menampilkan data employee digunakan perintah:

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

Hasil perintah tersebut ditunjukkan dalam Gambar 11.

```
mysql> select emp_id, emp_name, address, gender, position from tb_employee;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | address | gender | position |
+-----+-----+-----+-----+-----+
| 501234 | Andwel | Jl. Mawpati 53 | F | Waitres |
| 500500 | Kurosaki ichigo | Jl. Serenetai | M | Sales |
| 505566 | Lulu | Jl. Ditempat 123 | F | Admin |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Gambar 11. Data Tabel *tb\_employee*

**3) Tabel tb\_counter**

Tabel *tb\_counter* digunakan untuk menyimpan data counter makanan yang terdapat pada foodcourt tersebut.

Untuk menampilkan data counter makanan digunakan perintah:

```
SELECT counter_id, counter_name, address, company_name, share_amt FROM tb_counter.
```

Hasil perintah tersebut ditunjukkan dalam Gambar 12.

```
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 A11 Baba | Jl. Maju 3 | Ali Baba Corp. | 10 |
| S02 | Big Burger | Jl. Pelan 66 | Pi. Semu Suka | 10 |
| S03 | Fiesta Steak | Jl. Molina 32 | Fiesta | 15 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Gambar 12. Data Tabel *tb\_counter*

**4) Tabel tb\_food**

Tabel *tb\_food* digunakan untuk menyimpan data makanan yang disediakan oleh setiap counter makanan yang terdapat pada foodcourt tersebut.

Untuk menampilkan data counter makanan digunakan perintah:

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

Hasil perintah tersebut ditunjukkan dalam Gambar 13.

```
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 13. Data Tabel *tb\_counter*

**5) Tabel tb\_user**

Tabel *tb\_user* digunakan untuk menyimpan data user yang berhak untuk mengakses Sistem Transaksi Pembayaran Food Court.

Untuk menampilkan data customer digunakan perintah:

```
SELECT user_id, counter_name, emp_name FROM tb_user;
```

Hasil perintah tersebut ditunjukkan dalam Gambar 14.

```
mysql> select user_id, password, counter_id, emp_id
-> from tb_user
-> group by user_id asc;
```

user_id	password	counter_id	emp_id
counter1	counter1	s01	505566
counter2	counter2	s02	508899
counter3	counter3	s03	506650
sysadmin	test	admin	sysadmin

4 rows in set (0.00 sec)

Gambar 14. Data Tabel *tb\_user*

#### 6) Tabel *tb\_trans\_dtl*

Tabel *tb\_trans\_dtl* digunakan untuk menyimpan data lengkap mengenai order makanan yang dilakukan oleh customer.

Untuk menampilkan data order makanan digunakan perintah:

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

Hasil perintah tersebut ditunjukkan dalam Gambar 15.

```
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 15. Data Tabel *tb\_trans\_dtl*

#### 7) Tabel *tb\_trans\_hist*

Tabel *tb\_trans\_hist* digunakan untuk menyimpan data jenis transaksi yang dilakukan oleh customer.

Untuk menampilkan data transaksi yang telah dilakukan customer digunakan perintah:

```
SELECT trans_date, trans_type, customer_id,
counter_id, ref_no, amount
FROM tb_trans_hist;
```

Hasil perintah tersebut ditunjukkan dalam Gambar 16.

```
mysql> select trans_date, trans_type, customer_id, counter_id, ref_no, amount
-> from tb_trans_hist;
```

trans_date	trans_type	customer_id	counter_id	ref_no	amount
2008-10-10 22:23:34	PAYMENT	0809140001	s01	E	15000
2008-10-10 22:24:04	PAYMENT	0809140003	s01	1	10000
2008-10-10 22:32:20	PAYMENT	0809140001	s01	2	20000
2008-10-10 23:01:05	PAYMENT	0809140001	s01	3	4000
2008-10-13 10:38:36	NEW CREDIT	0809010004	-	-	100000

5 rows in set (0.03 sec)

Gambar 16. Data Tabel *tb\_trans\_hist*

#### 8) Tabel *tb\_trans\_mstr*

Tabel *tb\_trans\_mstr* digunakan untuk menyimpan rekapan data transaksi customer yang diambil dari tabel *tb\_trans\_dtl* dan *tb\_trans\_hist*.

```
SELECT trans_id, counter_id, date, emp_id,
customer_id, desk_no
FROM tb_trans_mstr;
```

Hasil perintah tersebut ditunjukkan dalam Gambar 17.

```
mysql> select trans_id, counter_id, date, emp_id, customer_id, desk_no
-> from tb_trans_mstr;
```

trans_id	counter_id	date	emp_id	customer_id	desk_no
0	s01	2008-10-10 22:23:34	counter1	0809140001	5
1	s01	2008-10-10 22:24:04	counter1	0809140003	4
2	s01	2008-10-10 22:32:20	counter1	0809140001	4
3	s01	2008-10-10 23:01:05	counter1	0809140001	12

4 rows in set (0.00 sec)

Gambar 17. Data Tabel *tb\_trans\_mstr*

### C. Pengujian Pembacaan Barcode pada E-card

#### • Prosedur Pengujian

TABEL 3. DATA PENGUJIAN PEMBACAAN BARCODE

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

Data hasil pengujian pembacaan barcode berdasarkan jarak dan sudut kemiringan tertentu ditunjukkan pada Tabel 4.

TABEL 4. DATA HASIL PENGUJIAN PEMBACAAN BARCODE

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

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°.

## VI. KESIMPULAN

Berdasarkan hasil perancangan dan implementasi serta pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

- Hasil pengujian koneksi basis data menunjukkan bahwa koneksi *server* basis data MySQL dan Sistem Transaksi Pembayaran *Food Court* melalui sebuah jaringan LAN (*Local Area Network*) dapat bekerja dengan baik.
- Aplikasi Admin dan Operator *Counter* menunjukkan bahwa sistem dapat diimplementasikan sesuai analisis kebutuhan yang telah dilakukan.
- Proses pembacaan/*scanning barcode* menunjukkan bahwa *barcode* dapat terbaca dengan baik antara jarak 0 – 2 cm.
- Hasil dari seluruh rangkaian pengujian menunjukkan bahwa sistem yang dikembangkan adalah valid sesuai kebutuhan.

## DAFTAR PUSTAKA

- [1] [SUW-91] Suwardjono : "Akuntansi Pengantar", BPFE Yogyakarta, 1991.
- [2] [NUG-04] Nugroho, Bunafit : "Database Relasional dengan MySQL", Andi Yogyakarta, 2004.
- [3] [WAG-05] Wagito : "Jaringan Komputer Teori dan Implementasi Berbasis Linux", Gava Media, 2005.
- [4] [FAT-05] Fathansyah : "Basis Data", Informatika Bandung, 2005
- [5] [KUS-03] Kusumo, Ario S : " Pemrograman Database dengan Visual Basic 6.0", Elex Media Komputindo, 2003.
- [6] [SUP-06] Supardi, Yuniar : "Implementasi Desain Database dalam VB 6.0 dan MySQL", Elex Media Komputindo, 2006.
- [7] [BAR-06] [http://barcode.com/how\\_barcodes\\_are\\_read.shtml.htm](http://barcode.com/how_barcodes_are_read.shtml.htm)
- [8] [HAT-04] [http://ilkom.unsri.ac.id/dosen/hartini/materi/VIII\\_DFD.pdf](http://ilkom.unsri.ac.id/dosen/hartini/materi/VIII_DFD.pdf)
- [9] [HID-03] Hidayat, Rahmat, 2003, Belajar Database MySQL, <http://ilmukomputer.com/umum/rahmat/mysql.php>
- [10] [IRM-03] Irmansyah, Faried, 2003, Pengantar Database, <http://ilmukomputer.com/umum/...>
- [11] [MSD-07] [http://msdn2.microsoft.com/en-us/library/aa231216\(VS.60\).aspx](http://msdn2.microsoft.com/en-us/library/aa231216(VS.60).aspx)
- [12] [SEM-00] <http://semiconsystems.com>