BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

1. Konsep Dasar Sistem Informasi

Menurut Japerson (2014:2) Sistem adalah "suatu jaringan kerja dari prosedur - prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegaiatan atau sasaran yang tertentu".

Sedangkan pendekatan yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Menurut Richard F. Neuschel dalam Japerson (2014:3) bahwa "Suatu prosedur adalah suatu urutan operasi klerikal (tulis-menulis), yang melibatkan beberapa orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

Menurut Japerson (2014:9) mengatakan "Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang penerimanya". Sedangkan menurut Gordon B. Davis dalam Japerson (2014:9) Informasi adalah "Data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

Menurut Azahra sutanto dalam Puspitawati dkk. (2011:15) mengemukakan bahwa "sistem informasi merupakan komponen-komponen dari subsistem yang saling berhubungan dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi.".

Menurut Burch dan Grudnitski dalam Puspitawati dkk. (2011:20) mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu:

1. Blok Masukan (*Input Block*)

   Input yang mewakili data yang masuk ke dalam sistem informasi. Input disini dapat termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

   Terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan dalam basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

   Teknologi merupakan “kotak alat” dalam sistem informasi yang digunakan untuk menerima input, menjadikan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran serta membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

   Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

   Beberapa pengendalian perlu dirancang untuk mencegah kerusakan, kegagalan sistem yang mungkin terjadi.

2. Konsep Dasar Model Pengembangan Sistem

Menurut Rosa dan M. Shalahuddin (2013:28) menjelaskan bahwa "model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)". Ada 5 tahapan dalam metode ini antara lain: analisis, desain, pembuatan kode program, pengujian, dan pemeliharaan.

Berikut adalah tahapan dalam pengembangan perangkat lunak dengan metode *waterfall*:

1. Analisis

   Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

   Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data,

arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

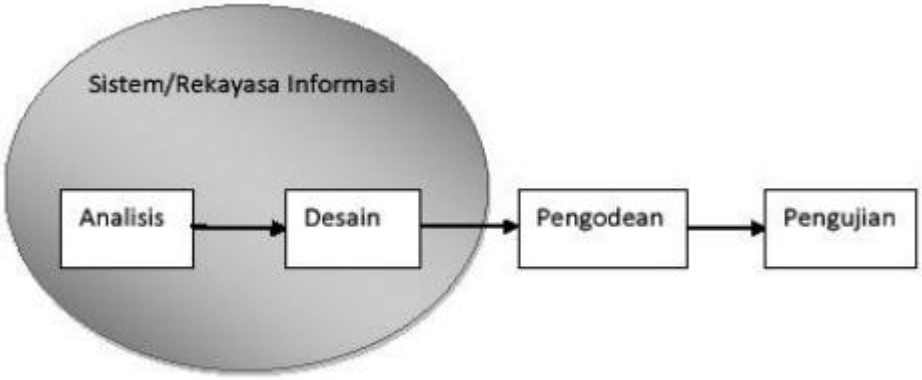
   Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

   Pengujian fokus pada perangkat lunak secara segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pemeliharaan (*Maintenance*)

   Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari tahap analisis spesifikasi untuk perubahan perangkat lunak baru.



Sumber: Rosa dan M. Shalahuddin (2013:29)

Gambar II.1.
Model *Waterfall*

Kemunculan model air terjun adalah untuk membantu mengatasi kerumitan yang terjadi dari proyek-proyek pengembangan perangkat lunak, sebuah model air terjun untuk memperinci apa yang seharusnya perangkat lunak lakukan (mengumpulkan dan menentukan kebutuhan sistem) sebelum sistem dikembangkan. Model ini memungkinkan pemecahan misi pengembangan yang rumit menjadi beberapa langkah logis dengan beberapa langkah yang pada akhirnya akan menjadi produk akhir yang siap pakai. (Simarmata, 2010:54).

3. Konsep Dasar Pemrograman

Menurut Sugiyono (2005:14) dijelaskan bahwa "Definisi pemrograman terstruktur adalah merupakan suatu tindakan atau metode untuk mengorganisasikan dan membuat kode-kode program supaya mudah untuk dimengerti, mudah di *test* dan mudah dimodifikasi".

Ada beberapa hal yang menjadi tujuan dilakukan pemrograman terstruktur, yaitu:

1. Meningkatkan kehandalan program
2. Program mudah dibaca dan dapat ditelusuri apabila ada kesalahan-kesalahan
3. Menyederhanakan dari kerumitan program
4. Menyederhakan penulisan program
5. Meningkatkan kualitas dan produktivitas program

Dalam pembuatan pemrograman terstruktur ada beberapa kriteria yang harus dipenuhi sehingga suatu program itu bisa disebut pemrograman terstruktur, diantaranya:

1. Ekspresifitas

Bahasa pemrograman yang baik harus jelas dalam menggambarkan algoritma yang dibuat.

2. Definitas

Bahasa pemrograman dapat didefinisikan dari adanya sintak dan semantik serta bahasa pemrograman yang baik haruslah konsisten dan tidak bermakna ganda

3. Tipe data dan strukturnya

Bahasa pemrograman yang baik harus berkemampuan dalam mendukung berbagai tipe data (*Integer, Real, String, Boolean* dan lain sebagainya) serta struktur data (*Array, Record, File* dan sebagainya).

4. Modularitas

Bahasa pemrograman yang baik harus mempunyai fasilitas subprogram, sehingga suatu program yang besar dapat dikerjakan oleh beberapa pemrogram

secara bersama-sama yang nantinya dengan mudah dapat digabungkan hanya menjadi sebuah modul saja.

5. Adanya *Input-Output*

Bahasa pemrograman yang baik harus dapat mendukung berbagai jenis model file seperti Sequensial, random, index dan sebagainya dalam proses masukan dan keluaran.

6. Portabilitas

Bahasa pemrograman yang dapat digunakan pada berbagai tipe mesin komputer yang berbeda-beda, sehingga dapat menjadi bersifat *machine independent.*

Adanya portabilitas suatu program ditentukan pada ketergantungan program tersebut terhadap mesin komputer atau sistem operasi, maka semakin banyak usaha yang diperlukan untuk memindahkan program tersebut, sehingga dengan adanya ketergantungan pada mesin akan sangat berdampak pada penggunaan piranti masukan dan keluaran, misalnya pada saat mengakses suatu berkas atau file.

7. Efisiensi

Bahasa pemrograman yang dapat mengatur banyaknya instruksi program dalam membatasi waktu tempuh pemrosesan, mengatur besar dan jenis input data yang digunakan serta jumlah memori yang digunakan program.

8. Interaktif

Bahasa pemrograman yang baik harus mudah dipelajari dan diajarkan pada *User* serta mudah dimengerti tentang proses yang sedang dilakukan oleh program.

9. Umum

Bahasa pemrograman yang baik harus memiliki jangkauan yang luas untuk berbagai aplikas pemrograman sehingga dapat bersifat bahasa serbaguna.

Pemrograman terstruktur harus mengikuti aturan dan teknik yang telah diberlakukan. Adanya beberapa cara atau teknik dalam pembuatan pemrograman terstruktur, diantaranya:

1. Pemrograman modular

Pemrograman modular biasanya menggunakan pernyataan *subroutine* yaitu kumpulan perintah yang melakukan tugas terbatas. Misalnya dalam bahasa Pascal dan menggunakan *procedure* dan *function.*

Kriteria program modular:

a. Program dipecah-pecah kedalam modul-modul
b. Setiap modul mempunyai tugas dan fungsi sendiri
c. Setiap modul ditulis secara terpisah dengan modul lainnya, shingga program mudah dicari kesalahannya.
d. Setiap program mempunyai modul program utama, untuk mengontrol semua proses submodul yang terjadi, termasuk mengirimkan kontrol program ke submodul untuk melakukan tugas dan fungsi tertentu.
e. Setiap submodul mengembalikan kontrol program ke modul utama apabila selesai melakukan tugasnya.

2. Pemrograman *top-down*

Mendefinisikan program modul utama yang pertama kali dijalankan, selanjutnya memanggil modul lainnya (submodul) untuk melakukan tugas dan

juga untuk mengakhiri proses program tersebut. Kriteria dari pemrograman *top-down* biasanya berbentuk diagram HIPO *(Hierarchy plus Input-Process-Output).*

4. *Unified Modelling Language* (UML)

Menurut Rosa (2013:13) "UML (*Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement,* membuat analisi dan design, serta menggambarkan arsitektur dalam pemrogramman berorientasi objek".

Abstraksi konsep dasar UML terdiri dari *structural classification, dynamic behaviour* dan *model management.*

Menurut Yulianta dan Mursanto (2010:115) "*Unified Modeling Language* (UML) digunakan sebagai notasi utama untuk mengambarkan dan mendokumentasikan sistem yang dibangun".

1. *Usecase* Diagram

Menurut Rosa (2013:155) "*Usecase* merupakan pemodelan untuk tingkah laku (*behavior*) sistem informasi yang akan dibuat". *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

2. *Activity* Diagram

Menurut Rosa (2013:161) "*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak".

Activity diagram mempunyai peran seperti halnya *flowchart,* akan tetapi perbedaan dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. *Activity* diagram merupakan *state diagram* khusus, dimana sebagian besar *state* sebelumnya (*internal processing).* Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum.

3. *Component* Diagram

Menurut Rosa (2013:148) "*Component* diagram dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem". Sebuah komponen bisa mengakses *service* tersebut disebut *export interface* sedangkan yang mengaksesnya disebut *import interface. Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency)* diantara komponen.

Komponen piranti lunak adalah model berisi kode, baik berisi *source code* maupun *binary code,* baik *library* maupun *executable,* baik yang muncul pada *compile time, link time,* maupun *run time.* Umumnya komponen terbentuk dari beberapa *class* dan atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface,* yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

4. *Deployment* Diagram

Menurut Rosa (2013:154) "*Deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi".

Deployment Diagram juga dapat digunakan untuk memodelkan hal-hal berikut:

a. Sistem tambahan (*embedded system)* yang menggambarkan rancangan *device node* dan *hardware.*
b. Sistem *client/server.*
c. Sistem terdistribusi.
d. Rekayasa ulang aplikasi.

Bagian *hardware* adalah *node* yaitu nama semua jenis sumber komputasi. Ada dua tipe note yaitu *processor* dan *device. Processor* adalah *node* yang bisa mengeksekusi sebuah komponen sedangkan *device* tidak. *Device* adalah perangkat keras seperti *printer*, *monitor* dan perangkat keras lainnya.

5. *Entity Relation Diagram* (ERD)

Entity Relation Diagram (ERD) atau yang dikenal juga dengan *Diagram Entity-Relationship* (Diagram E-R) merupakan suatu model yang menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

> Menurut Fatansyah (2007:79) mengemukakan bahwa *Model Entity-Relationship* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari dunia nyata yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Diagram Entity-Relationship*.

"Entity Relation Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas" (Simarmata dan Janner, 2010:67).

Notasi-notasi simbolik di dalam *Entity Relation Diagram* yang dapat digunakan adalah:

1. Entitas

   Digambarkan dengan kotak persegi panjang dan digunakan untuk menunjukkan sekumpulan orang, tempat, objek atau konsep dan sebagainya yang menunjukkan dimana data dicatat atau disimpan.

2. Hubungan atau Relasi

   Digambarkan dengan kotak berbentuk *diamond* atau belah ketupat dengan garis yang menghubungkan ke entitas yang terkait. Maka *relationship* diberi nama dengan kata kerja. Hubungan atau relasi menunjukkan abstraksi dari sekumpulan hubungan yang mengaitkan antara entitas yang berbeda.

3. Atribut

   Digambarkan dengan bentuk elips. Atribut menunjukkan karakteristik dari tiap entitas atau sesuatu yang menjelaskan entitas atau hubungan. Sehingga atribut dikatakan elemn dari entitas dan relasi. Dari setiap atribut entitas terdapat satu atribut yang dijadikan sebagai kunci (*key*). Beberapa jeni kunci tersebut antara lain : *Primary key, Candidate key, Composite key, Secondary key, Alternate key* dan *Foreign key*.

4. Tingkat Hubungan (*Cardinality*)

   Entity Relation Diagram (ERD) juga menunjukkan tingkat hubungan yang terjadi, dilihat dari segi kejadian atau banyak tiidaknya hubungan antara entitas tersebut, diantaranya ada tiga kemungkinan hubungan yang ada, yaitu:

a. Satu ke satu (*one to one* (1:1))

Tingkat hubungan dinyatakan satu ke satu jika suatu kejadian pada entitas pertama hanya mempunyai satu hubungan dengan satu kejadian maupun satu hubungan dengan satu kejadian pada entitas kedua. Demikian juga sebaliknya, satu kejadian pada entitas yang kedua hanya bisa mempunyai satu hubungann dengan satu kejadian pada entitas yang pertama.

b. Satu ke banyak (*one to many* (1:M))

Tingkat hubungan satu ke banyak (1:M) adalah sama dengan banyak ke satu (M:1), tergantung dari arah mana hubungan-hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian yang kedua, sebaliknya banyak kejadian pada entitas yang kedua hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. Banyak ke banyak (*many to many* (M:N))

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai hubungan dengan kejadian pada entitas lainnya, baik dilihat dari entitas yang pertama maupun dilihat dari entitas yang kedua.

6. *Logical Relationals Structures* (LRS) atau Model Relasional

Menurut Kusrini (2007:18), "Model relasional adalah kumpulan tabel-tabel untuk merepresentasikan data dan relasi antar data – data tersebut". Setiap tabel terdiri atas kolom-kolom dan setiap kolom mempunyai nama yang unik.

Logical Relationals Structures (LRS) dibentuk dengan nomor tipe *record*. Beberapa tipe record digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik.

2.2. Penelitian Terkait

Dalam proses pengolahan data dan penyajian menu perlu digunakannya suatu alat bantu dalam hal ini berupa aplikasi komputerisasi yang dapat mempermudah dan mempercepat sistem informasi pemesanan menu pada CAKI CAKE Karawang. Berikut beberapa tinjauan penelitian terdahulu (jurnal) yang dapat memperkuat alasan dibuatnya sistem informasi pemesanan menu hidangan pada CAKI CAKE Karawang, diantaranya:

> Menurut Christanto dkk. (2012:39) dalam jurnalnya bahwa:
> Pada era modernisasi, teknologi informasi memegang peranan yang sangat penting dalam kemajuan suatu foodcourt. Untuk memacu kemajuan tersebut teknologi informasi dapat dimanfaatkan guna mendapatkan suatu informasi yang cepat, tepat dan akurat. Oleh karena itu dibutuhkan suatu perangkat untuk meningkatkan suatu efisiensi dan produktivitas, sehubungan dengan hal tersebut maka pemakaian handphone tidak dapat dihindarkan lagi, berkaitan dengan keakuratan dan kecepatan pengolahan data sehingga dapat dihasilkan sistem informasi yang bermanfaat bagi foodcourt maupun pelanggan.

> Menurut Jevri Setia Nugroho dkk. (2014:127) dalam jurnalnya bahwa :
> Dengan memanfaatkan teknologi smartphone atau tablet yang saat ini sedang menjadi trend teknologi, pelaksanaan pemesanan konsumen di restoran dapat menjadi lebih teratur dan lebih akurat, selain dapat menghemat kertas karena pemesanan menu dicatat secara digital.

BAB III

ANALISA SISTEM BERJALAN

3.1. Tinjauan Institusi/Perusahaan

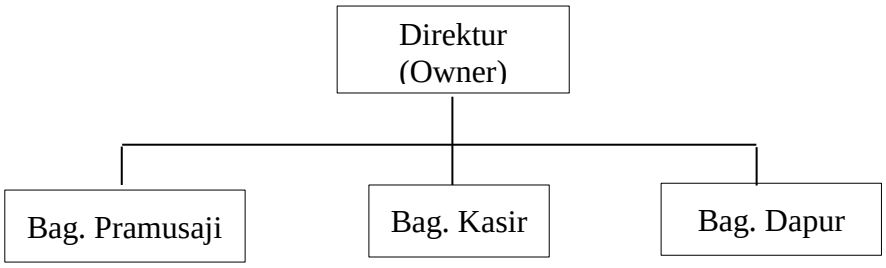
3.1.1. Sejarah Institusi/Perusahaan

Caki Cake pada awalnya menjual kue-kue rumahan yang didirikan oleh Ibu Restu Utami Dewi. Dengan sama sekali tidak ada latar belakang pendidikan di bidang kuliner, Alumnus S1 FIKOM UNPAD jurusan Public Relation dan Pasca Sarjana FE UNPAD Manajemen Magister ini memang gemar sekali membuat kue dan ia murni belajar dengan otodidak.

CakiCake berdiri pada akhir Oktober 2011 yang dimulai hanya sebagai bisnis rumahan. Asal mula pengambilan nama "CakiCake" ini diambil dari nama anaknya bernama Shaki yang biasa dipanggil Caki. Kemudian Cake sendiri berasal dari bahasa Inggris yang berarti Kue. Jadi kata "CakiCake" ini selain pengulangan katanya yang unik dan mudah diingat, CakiCake juga memiliki makna "Kue nya Caki". Kemudian ayahanda ibu Restu bapak Auh Solehudin membeli sebuah tempat usaha pada tahun 2012 yang diperuntukan untuk anak-anaknya menjual makanan dan kue keluarga. Sehingga *brand* tempat tersebut menggunakan nama caki cake yang dikelola oleh ibu Restu sampai tahun 2013, ditahun berikutnya tempat usaha tersebut dikelola oleh adik ibu Restu yaitu bapak Zia Firdaus sampai saat ini dan tidak merubah *brand* yang sudah banyak dikenal tersebut, akan tetapi produk yang dijualnya saat ini lebih kepada makanan dan minuman seperti halnya sebuah restoran berbeda dengan kakaknya yang hanya menjual kue.

3.1.2. Struktur Organisasi dan Fungsi

Struktur organisasi dan fungsi pada Restoran CAKI CAKE sebagai berikut :



Gambar III.1.
Struktur Organisasi

Adapun tugas dan fungsi dari setiap karyawan adalah sebagai berikut :

1. Direktur : Bertanggung jawab atas segala aktivitas operasional maupun keuangan, melakukan pengecekan atas laporan harian dan bulanan.
2. Bag. Pramusaji : Melayani dan mengantar makanan dari dapur kepada pengunjung.
3. Bag. Dapur : Mengolah dan menyiapkan makanan yang di pesan pengunjung.
4. Bag. Kasir : Bertugas untuk melayani kegiatan pembayaran pengunjung restoran, pemancingan dan waterboom.

3.2. Proses Bisnis Sistem

Berikut ini akan dijelaskan prosedur sistem berjalan pada restoran caki cake karawang yang telah dianalisa oleh penulis mempunyai tahapan-tahapan dalam melakukan proses kegiatannya. Pertama kali yang dilakukan dalam proses ini adalah *Customer* datang menduduki tempat yang tersedia dengan meminta daftar

menu hidangan, kemudian pramusaji memberikan daftar menu hidangan dan form pesanan. *Custumer* memilih menu hidangan dengan melihat daftar menu beserta harganya yang didapat dari pramusaji, setelah memilih menu, *customer* akan memberikan form pesanan yang kemudian diberikan kepada pramusaji. Pramusaji akan memberikan form pesanan tersebut kepada dapur untuk disiapkan menu-menu yang telah dipesan.

Pada bagian dapur setelah menyiapkan pesanan menu atas dasar dari bukti pesanan yang diterima dari pramusaji. Kemudian bukti pesanan diberikan kepada kasir sedangkan menu-menu tersebut diberikan kepada pramusaji untuk disajikan kepada *customer*.

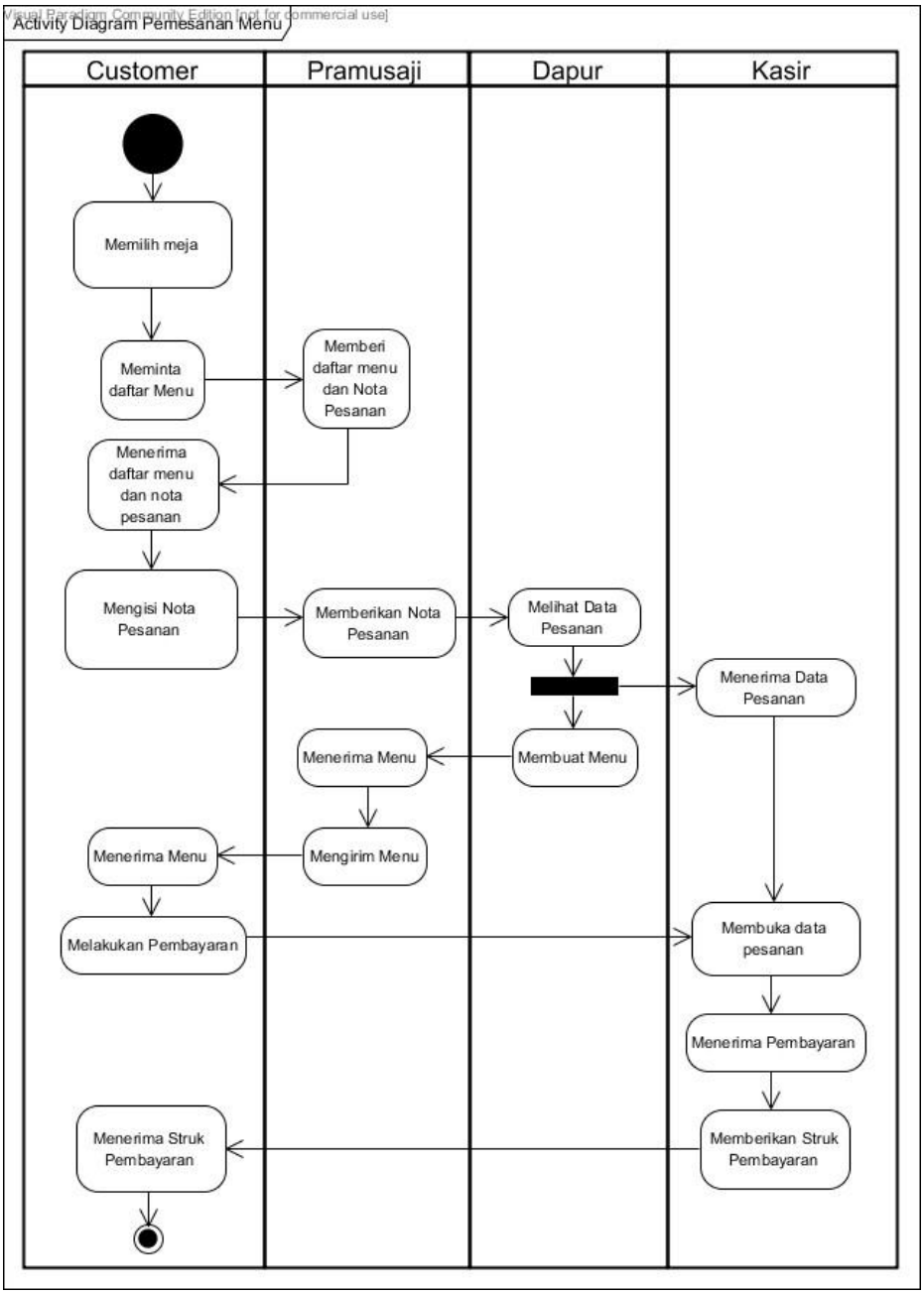
Pada proses selanjutnya *customer* melakukan pembayaran kepada kasir dengan menyebutkan nomor meja posisi *customer* menikmati hidangan kemudian kasir mencari data atas dasar bukti transaksi yang telah dibuat sebelumnya. Kemudian kasir memberikan struk pembayaran kepada *customer.*

Diagram *activity* memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat memodelkan sebuah alur kerja dari suatu aktivitas keaktivitas lainnya atau dari satu aktivitas ke keadaan sesaat (*state*). Diagram ini sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.

Pada bagian ini akan digambarkan logika prosedural, proses bisnis, dan jalur kerja sistem yang sedang berjalan pada Restoran CAKI CAKE Karawang, sehingga diharapkan dengan digambarkannya *activity* diagram ini akan mewakilkan gambaran alur proses sistem yang sedang berjalan. Diagram *activity*

yang dibahas pada proses bisnis sistem yang penulis akan paparkan antara lain, *activity Diagram* Pemesanan menu Restoran, *activity Diagram* Proses Penyajian Menu beserta *activity Diagram* Pembayaran Menu Restoran

1. *Activity Diagram* Pemesanan menu Restoran



Gambar III.2.

Activity Diagram Pemesanan Menu *Restaurant*

3.3. Spesifikasi Dokumen Sistem Berjalan

Spesifikasi sistem berjalan ini terdiri dari dua komponen yaitu dokumen masukan dan dokumen keluaran. Pada pemaparan kali ini akan dijelaskan beberapa dokumen, diantaranya dokumen masukan dan dokumen keluaran yang digunakan pada Restoran CAKI CAKE Karawang sebagai berikut :

1. Nama Dokumen : Nota Pesanan
   Fungsi : Digunakan sebagai mencatat pesanan menu di Restoran CAKI CAKE.
   Sumber : *Customer*
   Tujuan : Pramusaji
   Media : Kertas
   Frekuensi : Setiap kali adanya pesanan menu baru.
   Format : Lampiran A.1
2. Nama Dokumen : Daftar Menu
   Fungsi : Untuk melihat daftar Menu dan harganya.
   Sumber : Pramusaji
   Tujuan : *Customer*
   Media : Kertas
   Frekuensi : Setiap kali ada transaksi pemesanan
   Format : Lampiran A.2
3. Nama Dokumen : Struk Pembayaran
   Fungsi : Sebagai tanda bukti transaksi
   Sumber : *Kasir*
   Tujuan : *Costumer*

Media : Kertas

Frekuensi : Setiap terjadi transaksi pembayaran

Format : Lampiran A.3

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1. Analisa Kebutuhan *Software*

A. Tahapan Analisa

Usulan Sistem Penjualan pada Restoran dibuat menjadi sistem informasi penjualan yang diakses melalui aplikasi penjualan. Berikut ini spesifikasi kebutuhan *(System requirement)* dari sistem informasi penjualan restoran:

Halaman Pramusaji:

A1. Pengunjung melihat data menu.

A2. Pengunjung dapat mengisi data pesanan.

A3. Pramusaji dapat mencetak data pesanan.

Halaman Admin:

B1. Admin dapat mengelola data admin.

B2. Admin dapat mengelola data menu.

B3. Admin dapat mengelola data kategori.

B4. Admin dapat mengelola data transaksi.

B5. Admin dapat mengelola data laporan.

Halaman Dapur:

C1. Dapur dapat melihat data pesanan.

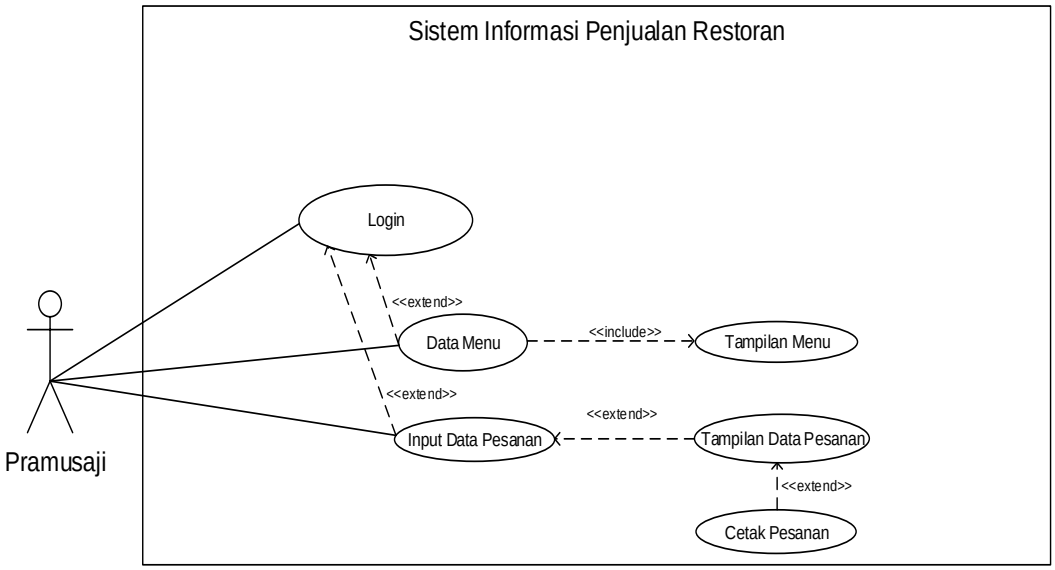
C2. Dapur dapat mencetak data pesanan.

C3. Dapur dapat melakukan konfirmasi pesanan.

B. *Use Case Diagram*

Use Case Diagram yang digunakan dalam rancangan web usulan adalah sebagai berikut :

1. *Use case* diagram Sisfo Penjualan Restoran halaman pramusaji



Gambar IV.1
Use case diagram Pramusaji

Deskripsi *Use case* Diagram Pemesanan Menu Halaman Pramusaji

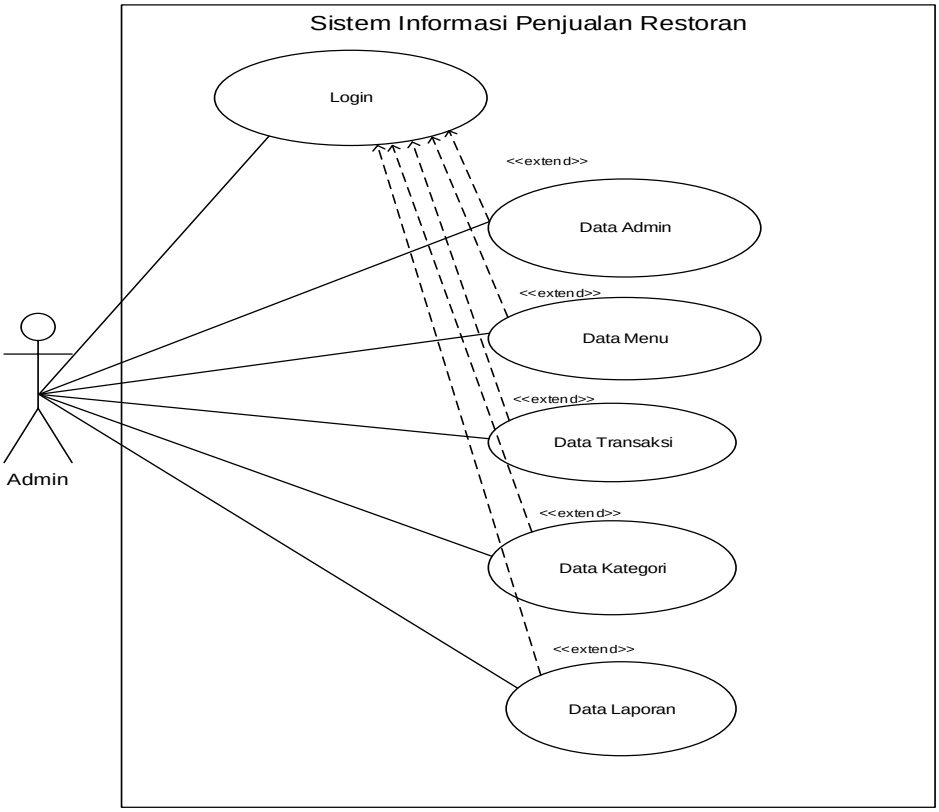
Tabel IV.1

Deskripsi *Use case* Diagram Pemesanan Menu Halaman Pramusaji

Use Case Name	Pemesanan Menu
Requirement	A1-A3
Goal	Pengunjung dapat melihat menu, memilih menu Pramusaji mecetak data pesanan
Pre-conditions	Pramusaji telah login
Post- conditions	Menu dapat terlihat. Menu dapat terinput. Data pesanan menu tercetak.
Failed.end conditions	Gagal melihat menu. Gagal terinput. Gagal tercetak.

Primary action	Pramusaji
Main flow / basic path	1. Pengunjung melihat menu 2. Pengunjung memilih menu 3. *System* menampilkan data menu 4. Pramusaji melakukan konfirmasi dengan memilih tombol "Lanjutkan" 5. *System* menampilkan form konfirmasi 6. Pramusaji menginput data konfirmasi 7. Pramusaji mencetak bukti pesanan menu 8. *System* menyimpan pesanan menu

2. *Use case* diagram Sisfo Penjualan Restoran halaman admin



Gambar IV.2
Use case diagram halaman Admin

a. Deskripsi *Use case* Mengelola Data Admin

Tabel IV.2

Deskripsi *Use case* Mengelola Data Admin

Use Case Name	Mengelola Data Admin
Requirement	B1
Goal	Admin dapat menambahkan, menghapus dan mengedit data admin
Pre-conditions	Admin telah login
Post- conditions	Data admin tersimpan, terupdate dan terhapus
Failed.end conditions	Gagal menyimpan, mngupdate dan menghapus
Primary action	Administrator
Main flow / basic path	1. Admin Melihat Data Admin & Form Admin 2. Admin menginput data admin baru pada form admin 3. Admin memilih tombol “simpan” 4. *System* menyimpan data admin
Alternate flow / Variant 1	A1. Admin memilih data admin A2. Admin memilih tombol *edit* A3. *System* menampilkan form data admin A4. Admin mengedit data admin
Invariant 2	B1. Admin memilih data admin B2. Admin memilih tombol hapus B3. *System* menghapus data admin

b. Deskripsi *Use case* Mengelola Data Menu

Tabel IV.3

Deskripsi *Use case* Mengelola Data Menu

Use Case Name	Mengelola data menu

Requirement	B2
Goal	Admin dapat menambahkan, menghapus dan mengedit data menu
Pre-conditions	Admin telah login
Post- conditions	Data menu tersimpan, terupdate dan terhapus
Failed.end conditions	Gagal menyimpan, mngupdate dan menghapus
Primary action	Administrator
Main flow / basic path	1. Admin Melihat Data Menu & Form Menu 2. Admin menginput data menu baru pada form menu 3. Admin memilih tombol "simpan" 4. *System* menyimpan data menu
Alternate flow / Variant 1	A1. Admin memilih data menu A2. Admin memilih tombol *edit* A3. *System* menampilkan form data menu A4. Admin mengedit data menu
Invariant 2	B1. Admin memilih data menu B2. Admin memilih tombol hapus B3. *System* menghapus data menu

c. Deskripsi *Use case* Mengelola Data Kategori

Tabel IV.4

Deskripsi *Use case* Mengelola Data Kategori

Use Case Name	Mengelola data kategori
Requirement	B3
Goal	Admin dapat menambahkan, menghapus dan mengedit data kategori
Pre-conditions	Admin telah login
Post- conditions	Data kategori tersimpan, terupdate dan terhapus
Failed.end conditions	Gagal menyimpan, mngupdate dan menghapus

Primary action	Administrator
Main flow / basic path	1. Admin Melihat Data Kategori & Form Kategori 2. Admin menginput data kategori baru pada form kategori 3. Admin memilih tombol "simpan" 4. *System* menyimpan data kategori
Alternate flow / Variant 1	A1. Admin memilih data kategori A2. Admin memilih tombol *edit* A3. *System* menampilkan form data kategori A4. Admin mengedit data kategori
Invariant 2	B1. Admin memilih data kategori B2. Admin memilih tombol hapus B3. *System* menghapus data kategori

d. Deskripsi *Use case* Mengelola Data Transaksi

Tabel IV.5

Deskripsi *Use case* Mengelola Data Transaksi

Use Case Name	Mengelola Data Transaksi
Requirement	B4
Goal	Admin dapat melihat detail Transaksi, melakukan pembayaran, mencetak data transaksi
Pre-conditions	Admin telah login
Post- conditions	Berhasil melihat detail transaksi, berhasil melakukan pembayaran, dan mencetak data transaksi
Failed.end conditions	Gagal melihat detail transaksi, melakukan pembayaran, dan mencetak data transaksi
Primary action	Administrator

Main flow / basic path	1. Admin memilih dan Melihat Data transaksi 2. Admin memilih tombol "detail transaksi" 3. *System* menampilkan form konfirmasi 4. Admin menginput data form dan memilih tombol "konfirmasi" untuk pelunasan transaksi 5. *System* merubah status transaksi 6. Admin memilih tombol "cetak" untuk memberikan bukti transaksi pembayaran 7. Admin memilih urutan id transaksi untuk melihat detail transaksi
Alternate flow / Variant 1	A1. Admin memilih dan Melihat Data Transaksi A2. Admin memilih urutan id transaksi untuk melihat detail transaksi
Invariant 2	-

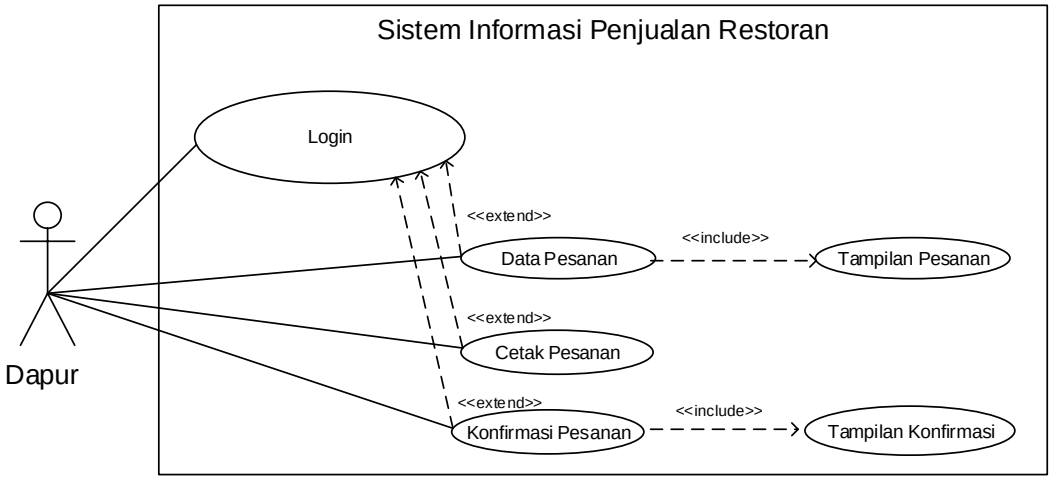
e. Deskripsi *Use case* Mengelola Data Laporan

Tabel IV.6

Deskripsi *Use case* Mengelola Data Laporan

Use Case Name	Mengelola Data Laporan
Requirement	B5
Goal	Admin dapat melihat laporan dan detail laporan
Pre-conditions	Admin telah login
Post- conditions	Dapat melihat laporan dan detail laporan
Failed.end conditions	Gagal melakukan data laporan dan detailnya
Primary action	Administrator
Main flow / basic path	1. Admin memilih dan Melihat Data laporan 2. Admin memilih periode transaksi untuk melihat detail transaksi
Alternate flow / Variant 1	-

3. *Use case* diagram berjalan *online* halaman Dapur



Gambar IV.3
Use case diagram halaman Dapur

a. Deskripsi *Use case* Melihat Data Pesanan

Tabel IV.7

Deskripsi *Use case* Melihat Data pesanan

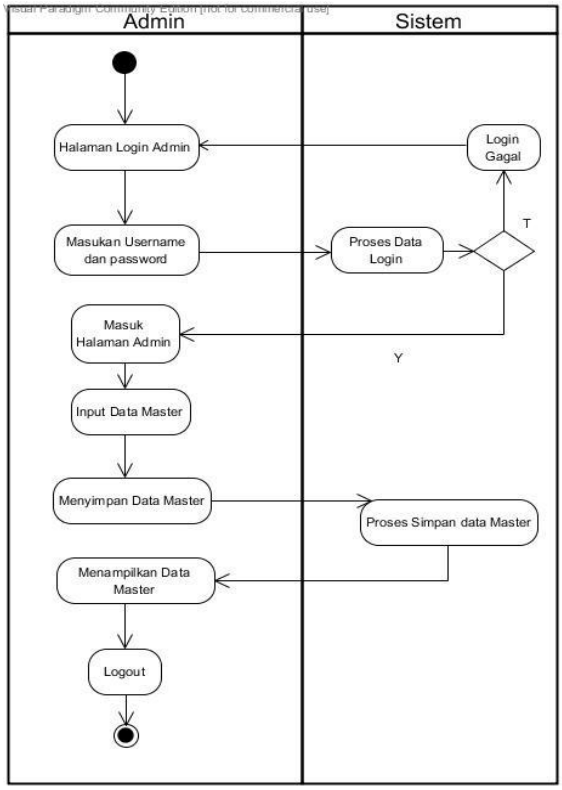
Use Case Name	Melihat Data pesanan
Requirement	C1-C3
Goal	Penyewa dapat melihat Data Pesanan, Mencetak data pesanan, Melakukan Konfirmasi
Pre-conditions	Admin Dapur telah login
Post- conditions	Data Pesanan dapat terlihat, Tercetak data pesanan, Melakukan Konfirmasi.
Failed end conditions	Gagal menampilkan Data Pesanan Gagal mencetak data pesanan Gagal melakukan konfirmasi
Primary action	Admin Dapur

Main flow / basic path	1. Admin Dapur memilih menu data Pesanan 2. *System* menampilkan Data Pesanan 3. Admin memilih tombol “Cetak” untuk bukti pesanan dapur 4. Admin memilih tombol “Pesanan Selesai” untuk konfirmasi bahwa menu telah di buat 5. *System* merubah status transaksi
Alternate flow / Variant 1	-

A. *Activity Diagram*

Activity Diagram yang digunakan dalam rancangan web usulan sebagai berikut :

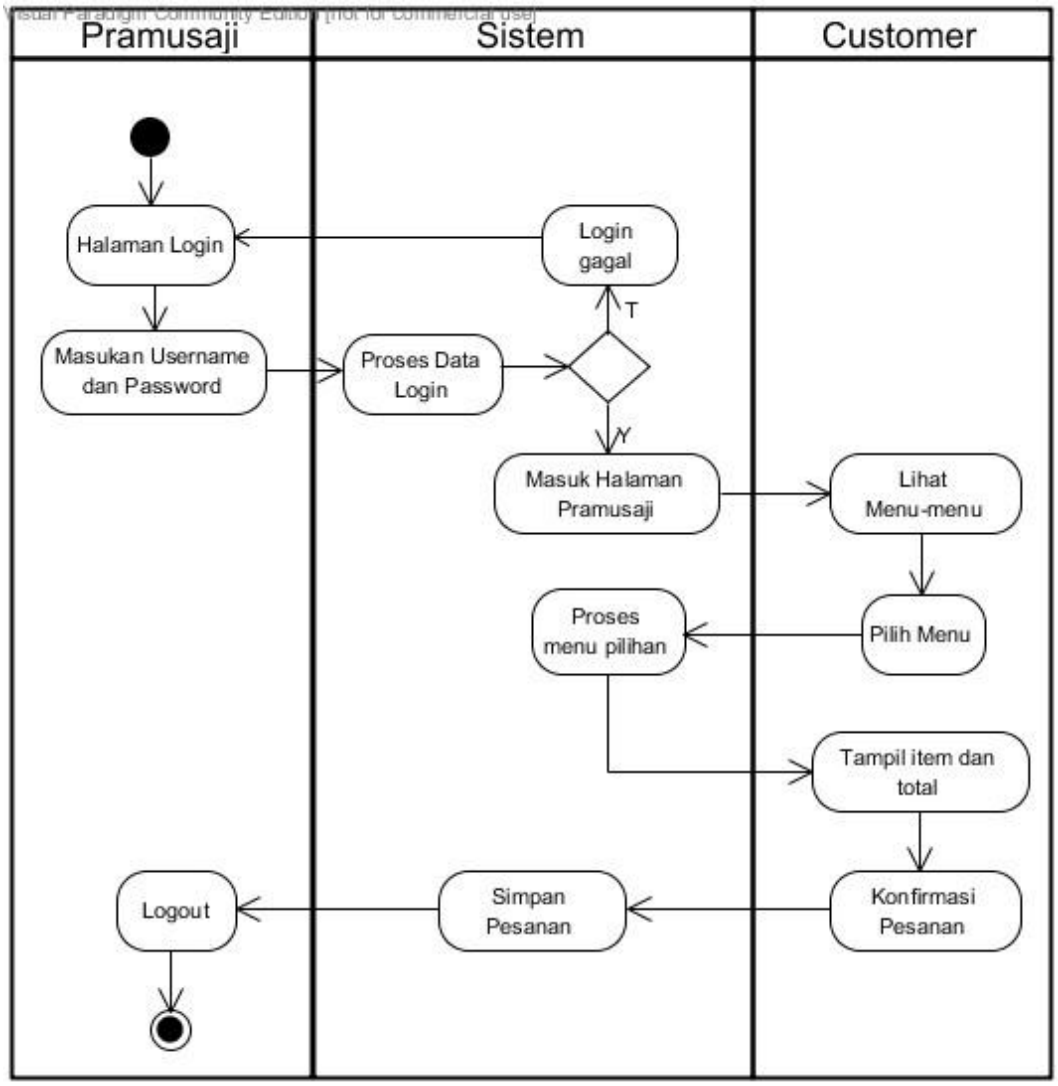
a. *Activity Diagram* Admin Penginputan Data Master



Gambar IV.4

Activity Diagram Admin Penginputan Data Master

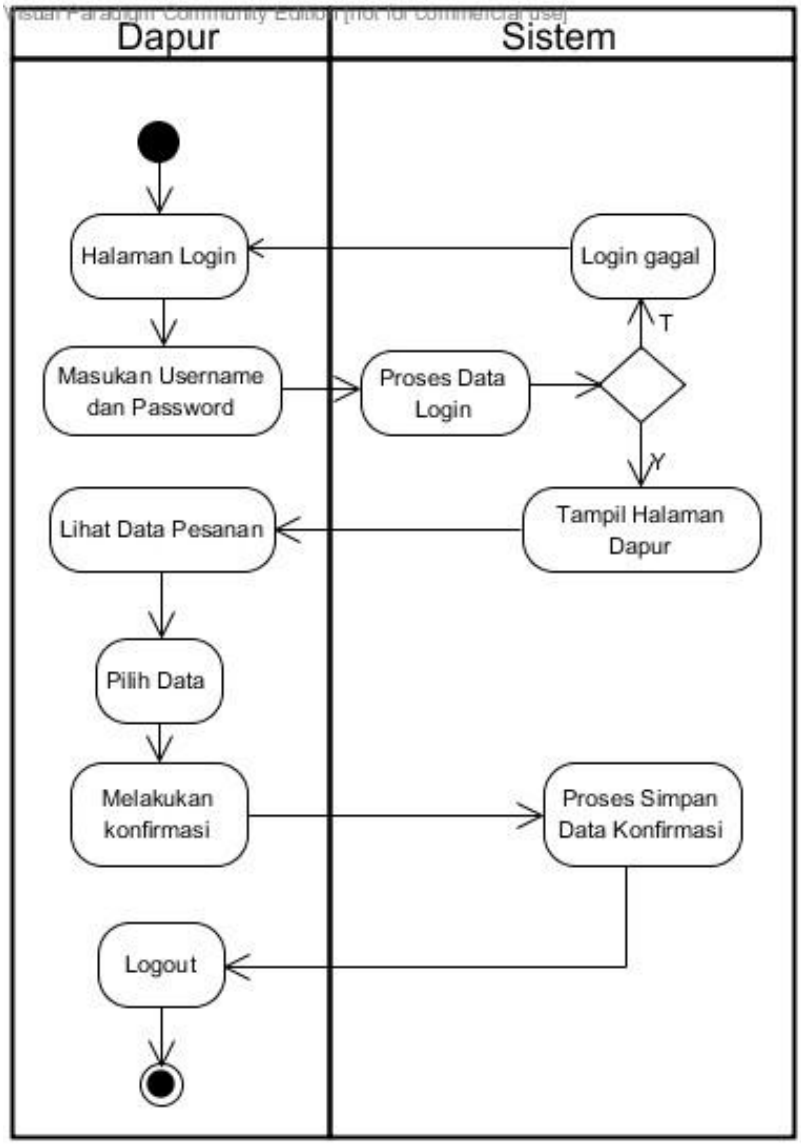
b. *Activity Diagram* Halaman Pramusaji Pemesanan



Gambar IV.5

Activity Diagram Pramusaji Pemesanan

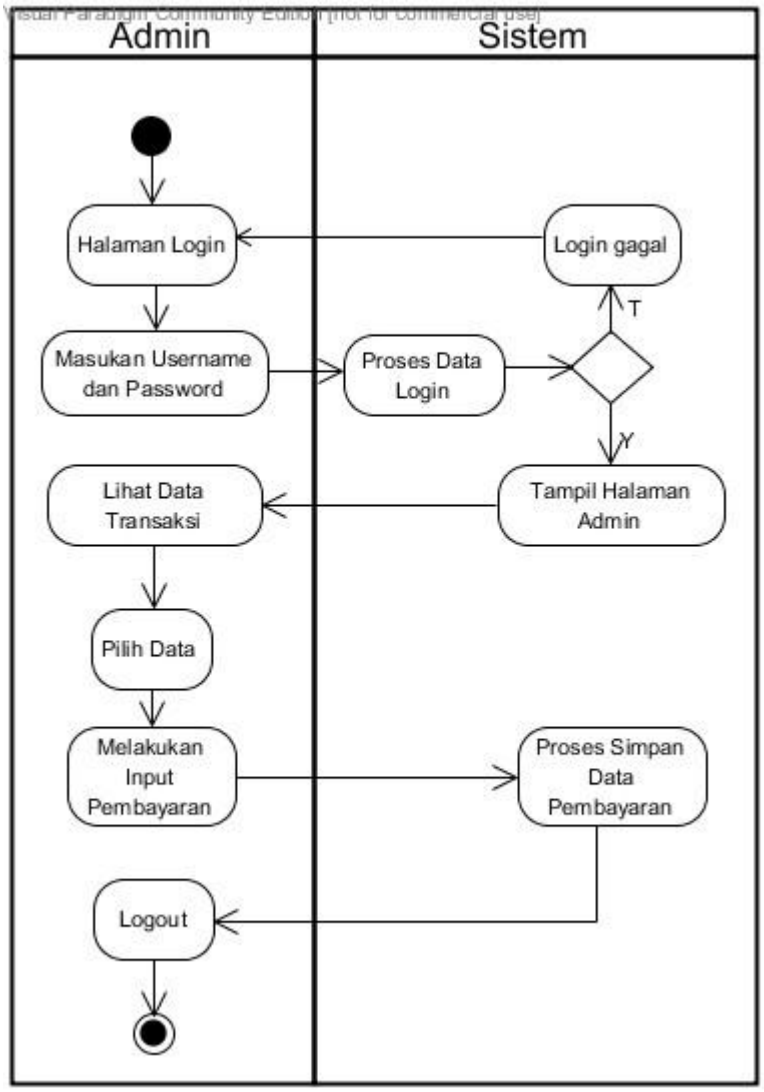
c. *Activity Diagram* Halaman Dapur



Gambar IV.6

Activity Diagram Halaman Dapur

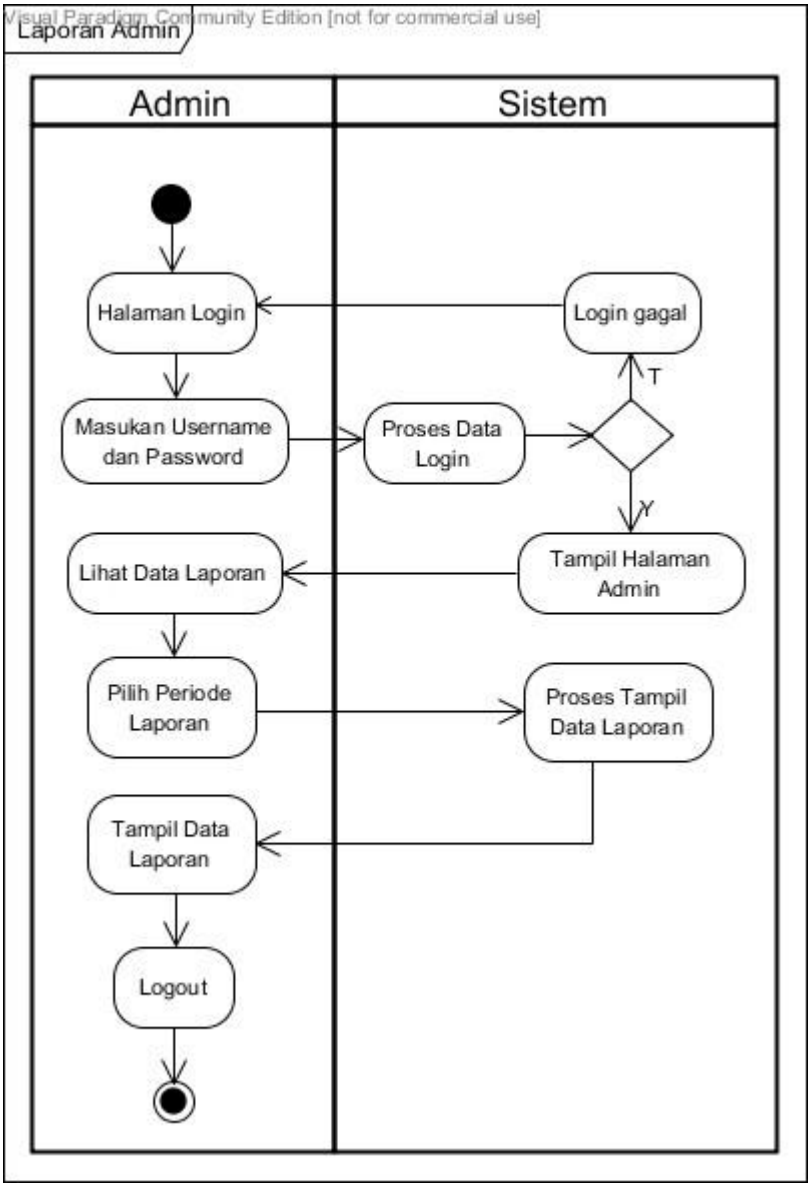
d. *Activity Diagram* Transaksi Admin



Gambar IV.7

Activity Diagram Transaksi Admin

e. *Activity Diagram* Laporan Admin



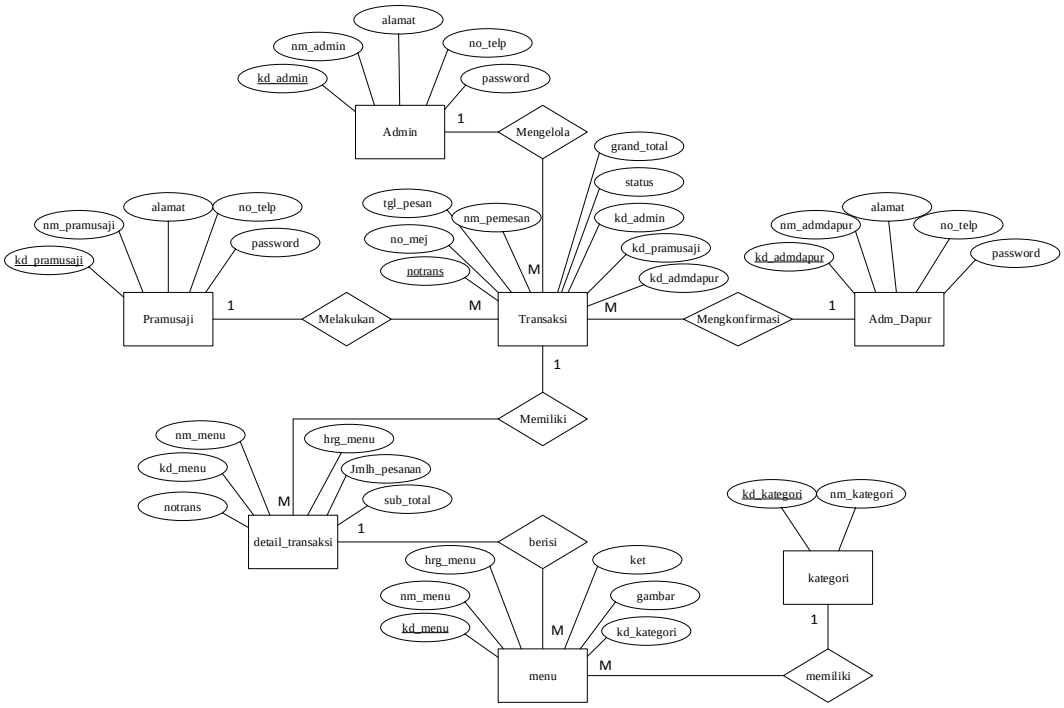
Gambar IV.8

Activity Diagram Laporan Admin

4.2. Desain

4.2.1. *Database*

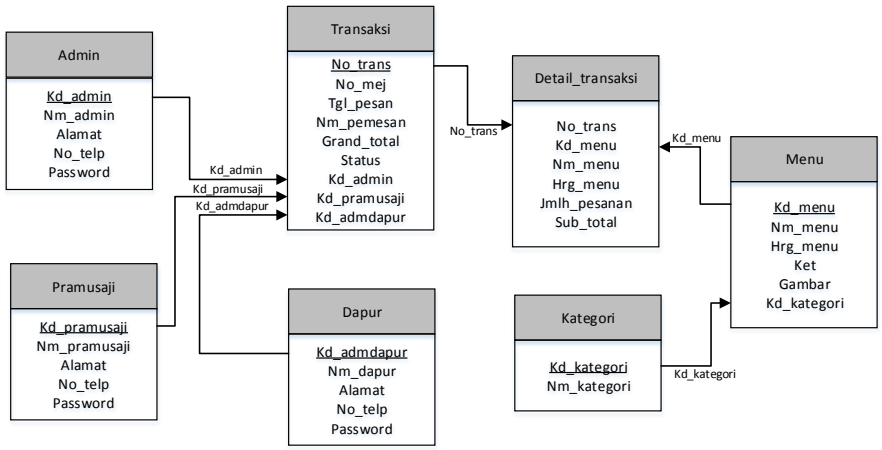
1. *Entity Relationship Diagram*



Gambar VI.9

Entity Relationship Diagram

2. *Logical Record Structure*



Gambar IV.10

Logical Record Structure Sistem Informasi Penjualan Restoran

3. Spesifikasi file

a. Tabel Admin

Tabel ini berfungsi untuk menyimpan data admin. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Admin

Akronim : admin

Tipe *File* : *File Master*

Akses *File* : *Random*

Panjang *Record* : 110

Kunci *Field* : kd_admin

Tabel IV.8

Spesifikasi tabel Admin

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	Kode Admin	kd_admin	*Varchar*	5	*Primary Key*
2	Nama Admin	nm_admin	*Varchar*	30	
3	Alamat	Alamat	*Text*		
4	No Telpon	no_telp	*Varchar*	15	
5	Password	Password	*Varchar*	60	

b. Tabel Pramusaji

Tabel ini berfungsi untuk menyimpan data admin. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Pramusaji

Akronim : pramusaji

Tipe *File* : *File Master*

Akses *File* : *Random*

Panjang *Record* : 110

Kunci *Field* : kd_pramusaji

Tabel IV.9

Spesifikasi tabel Pramusaji

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	Kode Pramusaji	kd_pramusaji	*Varchar*	5	*Primary Key*
2	Nama Pramusaji	nm_pramusaji	*Varchar*	30	
3	Alamat	Alamat	*Text*		
4	No Telpon	no_telp	*Varchar*	15	
5	Password	Password	*Varchar*	60	

c. Tabel Admin Dapur

Tabel ini berfungsi untuk menyimpan data admin. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Admin Dapur

Akronim : adm_dapur

Tipe *File* : *File Master*

Akses *File* : *Random*

Panjang *Record* : 110

Kunci *Field* : kd_admdapur

Tabel IV.10

Spesifikasi tabel admin dapur

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	Kode Admin Dapur	kd_admdapur	*Varchar*	5	*Primary Key*
2	Nama Admin Dapur	nm_pramusaji	*Varchar*	30	
3	Alamat	Alamat	*Text*		
4	No Telpon	no_telp	*Varchar*	15	
5	Password	Password	*Varchar*	60	

d. Tabel Menu

Tabel ini berfungsi untuk menyimpan data menu. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Menu

Akronim : menu

Tipe *File* : *File Master*

Akses *File* : *Random*

Panjang *Record* : 43

Kunci *Field* : kd_menu

Tabel IV.11

Spesifikasi tabel menu

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	Kode Menu	Kd_menu	*Varchar*	8	*Primary Key*
2	Nama Menu	Nm_menu	*Varchar*	30	
3	Harga	Hrg_menu	*Double*		
4	Keterangan	Ket	*Text*		

5	Gambar menu	Gambar	*Text*		
6	Kode Kategori	Kd_kategori	*Varchar*	5	*Foreign key*

e. Tabel Kategori

Tabel ini berfungsi untuk menyimpan kategori menu. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Kategori

Akronim : kategori

Tipe *File* : *File Master*

Akses *File* : *Random*

Panjang *Record* : 25

Kunci *Field* : kd_kategori

Tabel IV.12

Spesifikasi tabel kategori

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	Kode Kategori	Kd_kategori	*Varchar*	5	*Primary Key*
2	Nama Kategori	Nm_kategori	*Varchar*	20	

f. Tabel Transaksi

Tabel ini berfungsi untuk menyimpan data transaksi. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Transaksi

Akronim : transaksi

Tipe *File* : *File* Transaksi

Akses *File* : *Random*

Panjang *Record* : 66

Kunci *Field* : notrans

Tabel IV.13

Spesifikasi tabel Transaksi

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	No Transaksi	Notrans	*Varchar*	10	*Auto_Increment & Primary Key*
2	No Meja	No_mej	*Int*	3	
3	Tanggal Pesan	Tgl_pesan	*Datetime*	8	
4	Nama Pemesan	Nm_pemesan	*Varchar*	20	
5	Uang Bayar	Ubay	*Double*		
6	Uang Kembali	Ukem	*Double*		
7	Grand Total	Grand_total	*Double*		
8	Status	Status	*Varchar*	20	
9	Kode admin	Kd_admin	*Varchar*	5	*Foreign key*

e. Tabel Detail Transaksi

Tabel ini berfungsi untuk menyimpan data detail transaksi. Tabel ini terdiri dari *field-field* sebagai berikut:

Nama *Database* : restoran

Nama *File* : Tabel Detail Transaksi

Akronim : detailtransaksi

Tipe *File* : *File* Transaksi

Akses *File* : *Random*

Panjang *Record* : 59

Kunci *Field* : -

Tabel IV.14

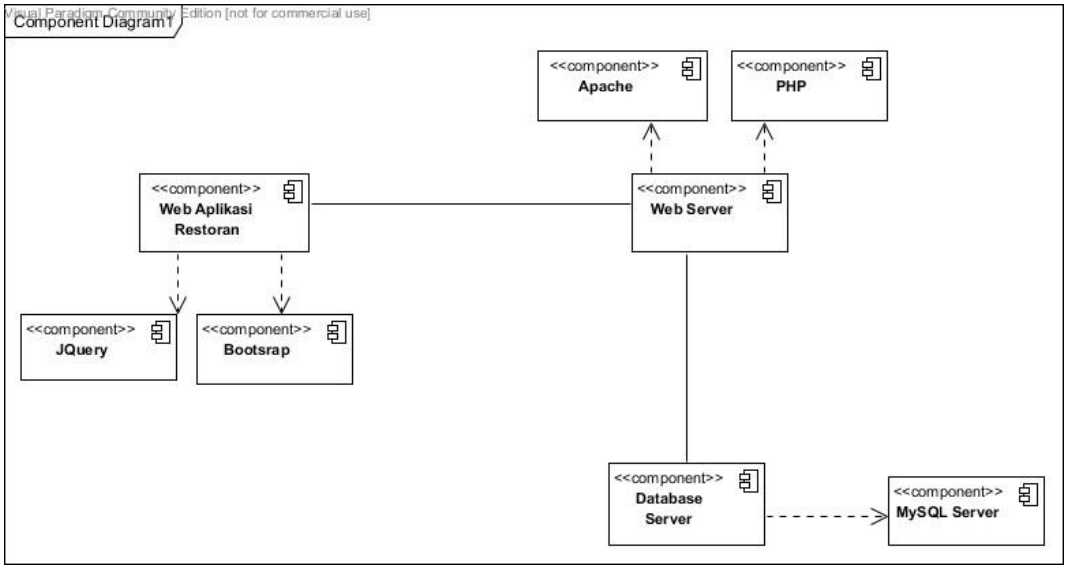
Spesifikasi tabel Detail Transaksi

No	Elemen Data	Akronim	*Type*	*Size*	Keterangan
1	No Transaksi	Notrans	*Varchar*	10	*Foreign key*
2	Kode Menu	Kd_menu	*Varchar*	8	*Foreign key*
3	Nama Menu	Nm_menu	*Varchar*	30	
4	Harga Menu	Hrg_menu	*Double*		
5	Jumlah pesanan	Jmlh_pesanan	*Int*	11	
6	Subtotal	Sub_total	*Double*		

4.2.2. *Software Architecture*

1. *Component Diagram*

Menggambarkan alokasi semua kelas dan obyek ke dalam komponen-komponen dalam desain fisik sistem *software*. *Diagram* ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software* seperti *source code, binary code* dan komponen tereksekusi.

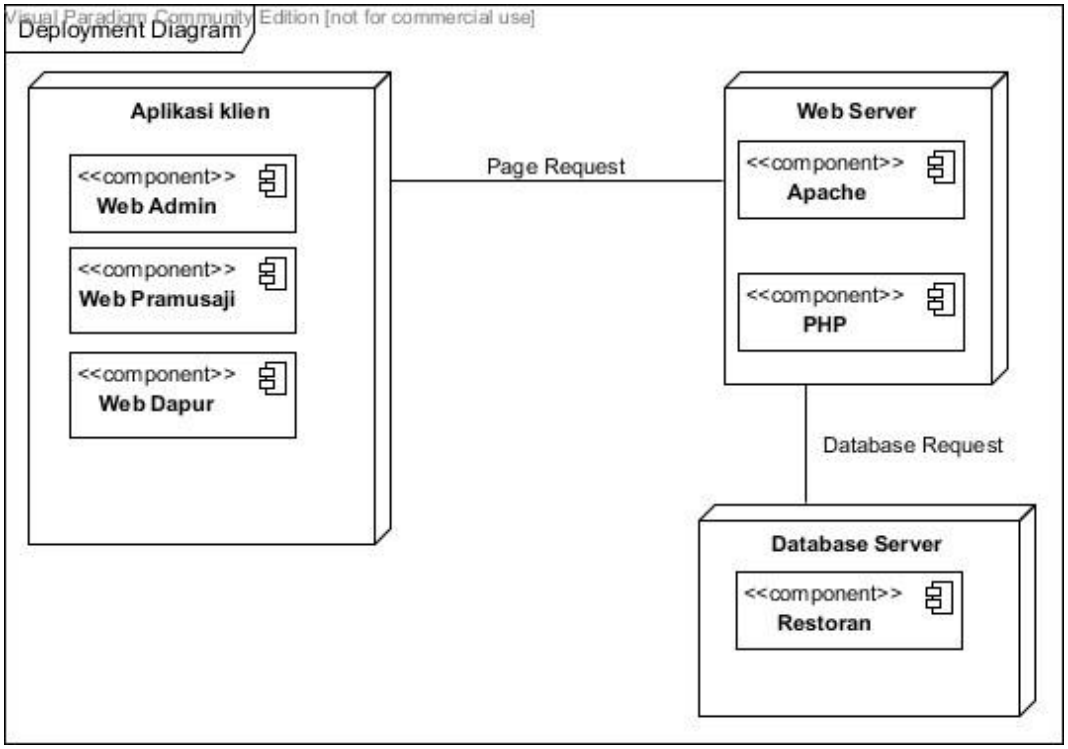


Gambar IV.11

Component Diagram

2. *Deployment Diagram*

Deployment diagram menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem diwakili oleh *node-node*, dimana masing-masing *node* diwakili oleh sebuah kubus. Garis yang menghubungkan kedua kubus menunjukkan hubungan diantara kedua *node* tersebut. Berikut gambar *Deployment Diagram :*

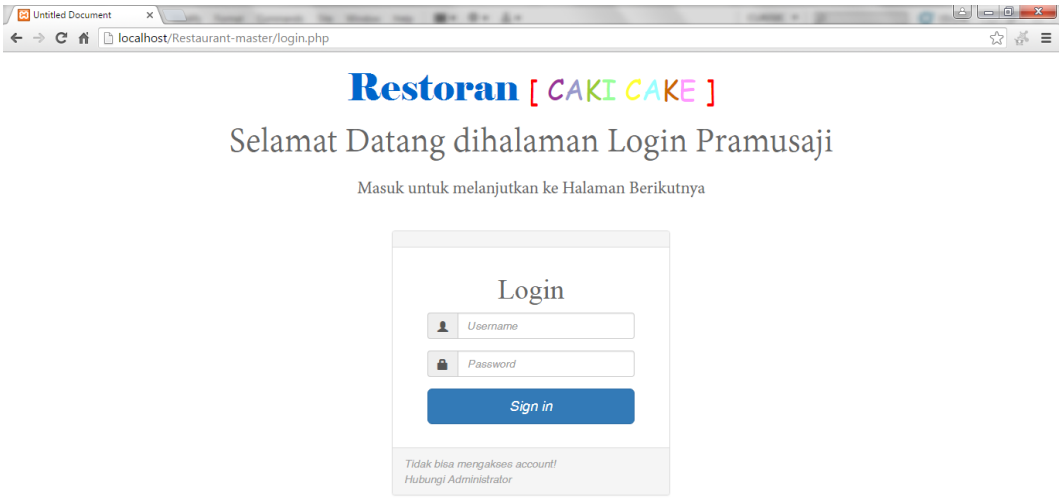


Gambar IV.12

Deployment Diagram

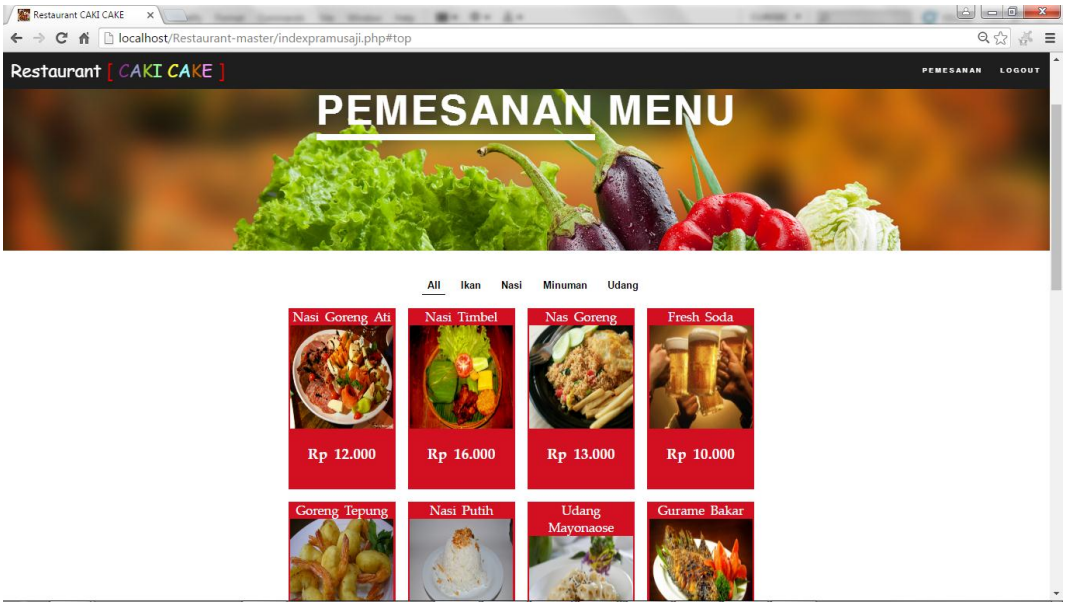
4.2.3. *User Interface*

Beberapa desain *user interface* sistem usulan yang ada pada website penjualan pada restoran caki cake, diantaranya:



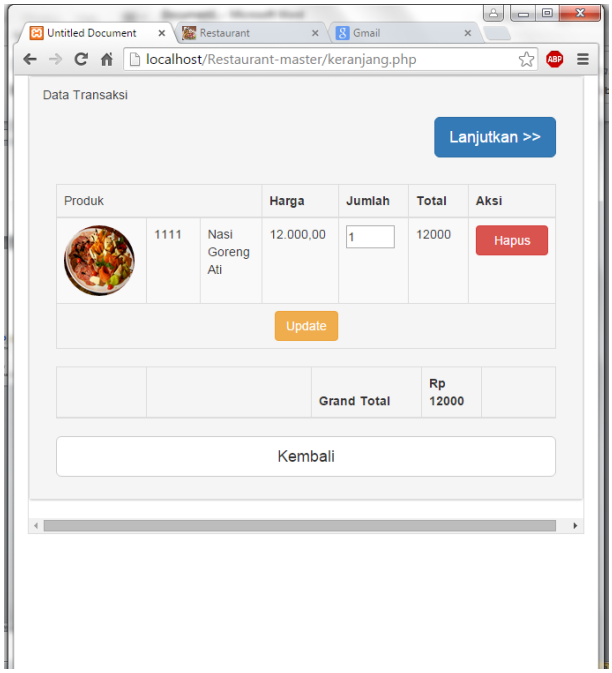
Gambar IV.13

Tampilan Login Pramusaji



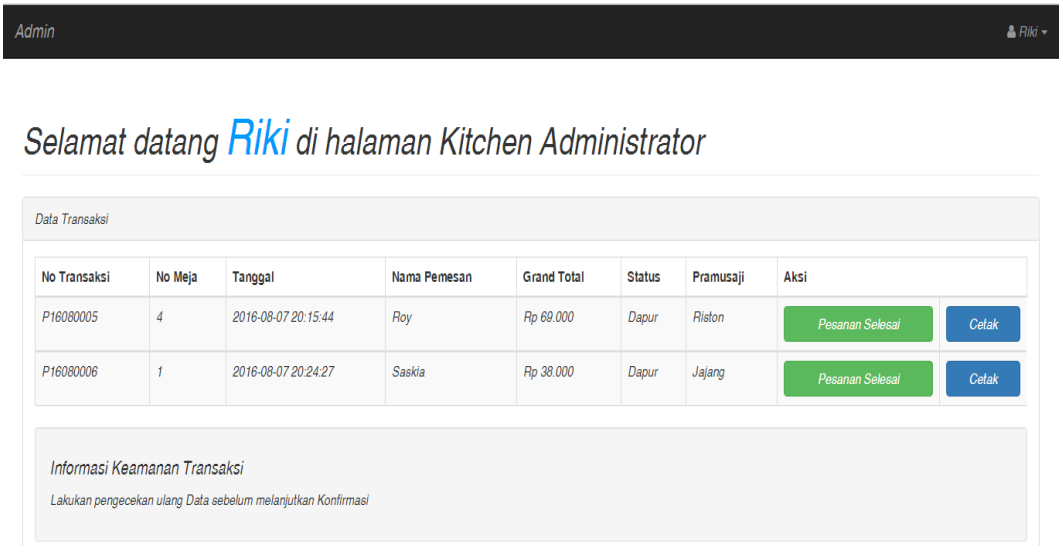
Gambar IV.14

Tampilan Halaman Pramusaji



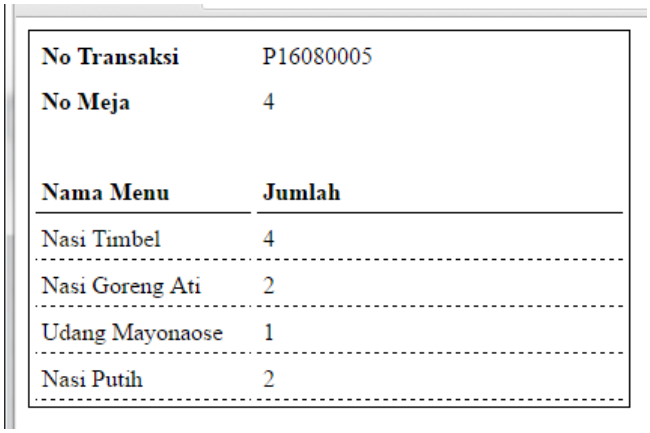
Gambar IV.15

Tampilan Halaman Data Pesanan



Gambar IV.16

Tampilan Halaman Dapur



No Transaksi	P16080005
No Meja	4

Nama Menu	**Jumlah**
Nasi Timbel	4
Nasi Goreng Ati	2
Udang Mayonaose	1
Nasi Putih	2

Gambar IV.17

Tampilan Cetak Pemesanan

4.3. Code Generation

A. Data Pesanan

```
<div class="container">
<div class="table-responsive">
<span class="center"></span>
<div class="row">
              <div class="col-lg-12">
                    <div class="panel panel-default">
                        <div class="panel-heading">
                            Data Transaksi
                            <div class="panel-body">
                            <!-- Button trigger modal -->
                                <div align="right">
                                <button class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">
                                Lanjutkan >>
                                </button>
                                </div>
                            </div>
                        <!-- /.panel-heading -->
                        <div class="panel-body">
                            <div class="dataTable_wrapper">
                              <form name="form2" method="post" action="updatepesanan.php">
                                <table width="80%" class="table table-striped table-bordered table-hover" id="dataTables-example">
                                  <?php
//$tampil=mysqli_query($koneksi,"select * from transaksi order by notrans desc");
//while($data=mysqli_fetch_array($tampil)){
?>
                                  <thead>
                                    <tr>
                                      <td colspan="3">Produk</td>
                                      <th width="12%">Harga</th>
                                      <th width="13%">Jumlah</th>
                                      <th width="11%">Total</th>
                                      <th>Aksi</th>
                                    </tr>
```

```
                                </tr>
                              </thead>
                              <?php
ini_set("display_errors",0);
$lihat=mysqli_query($koneksi,"select * from tmp_detail_trans where kd_admin='$kdadm'");
while($data=mysqli_fetch_array($lihat))
{
$kode=$data['kd_menu'];
$menu=mysqli_query($koneksi,"select * from menu where kd_menu='$kode'");
$row=mysqli_fetch_array($menu);
$total=$data['hrg_menu']*$data['jmlh_pesanan'];
$grand+=$total;
?>
                              <tbody>
                                <tr>
                                  <td width="11%" ><img src="New-folder/admin/gbrmenu/<?php echo $row['gambar'] ?>" width="80" height="80" class=
"img-circle"></td>
                                  <td width="10%"><?php echo $data['kd_menu'] ;?></td>
                                  <td width="11%"><?php echo $data['nm_menu'] ;?></td>
                                  <td><?php echo number_format($data['hrg_menu'],2,",",".")?></td>
                                  <td class="center"><input pattern="[0-9]{1,3}" title="Input hanya angka" type="number" name="tjml[]" id="tjml" size="3"
value="<?php echo $data['jmlh_pesanan']?>">
                                  <input name="harga[]" type="hidden" value="<?php echo $data['hrg_menu']?>"></td>
                                  <td class="center"><?php echo $total?>
                                  <td width="16%" class="center"><input name="kdmenu[]" type="hidden" value="<?php echo $data['kd_menu'] ;?>">
                                  <a onclick="return confirm ('Yakin Akan di hapus?')" href="hapuskeranjang.php?id=<?php echo $data['kd_menu']?>" class=
"btn btn btn-danger btn-block">Hapus</a></td>
                                </tr>
                                <?php } ?>
                                <tr>
                                  <td colspan="7" align="center"><input type="submit" name="button" id="button" value="Update" class="btn btn-warning"  ></
td>
                                      </tr>
                                    </tbody>
                                  </table>
                                </form>
                                <table width="102%" class="table table-striped table-bordered table-hover" id="dataTables-example2">
                                  <?php
//$tampil=mysqli_query($koneksi,"select * from transaksi order by notrans desc");
//while($data=mysqli_fetch_array($tampil)){
?>
                                <thead>
                                    <tr>
                                      <td width="18%"> </td>
                                      <th width="33%"> </th>
                                      <th width="22%">Grand Total</th>
                                      <th width="12%">Rp <?php echo $grand; ?></th>
                                      <th width="15%" colspan="3"> </th>
                                    </tr>
                                </thead>
```

B. Form Ruang Dapur

```
                                    <table class="table table-striped table-bordered table-hover" id="dataTables-example">
                                        <thead>
                                            <tr>
                                                <th>No Transaksi</th>
                                                <th>No Meja</th>
                                                <th>Tanggal</th>
                                                <th>Nama Pemesan</th>
                                                <th>Grand Total</th>
                                                <th>Status</th>
                                                <th>Pramusaji</th>
                                                <th colspan="4">Aksi</th>
                                            </tr>
                                        </thead>
<?php
$tgl = date('Y-m-d');
$tampil=mysqli_query($koneksi,"select * from transaksi where date(tgl_pesan)='$tgl' order by status, notrans asc");
while($data=mysqli_fetch_array($tampil)){
    $qry2=mysqli_query($koneksi,"select * from admin where kd_admin='$data[kd_admin]'");
    $data2=mysqli_fetch_array($qry2)

?>
                                        <tbody>
                                            <tr class="odd gradeX">
                                                <td><?php echo $data['notrans'];?></td>
                                                <td><?php echo $data['no_mej'];?></td>
                                                <td><?php echo $data['tgl_pesan'];?></td>
                                                <td class="center"><?php echo $data['nm_pemesan'];?></td>
                                                <td class="center"><?php echo "Rp "; echo number_format($data['grand_total'],0,",",".");?></td>
                                                <td class="center"><?php echo $data['status'];?></td>
                                                <td class="center"><?php echo $data2['nm_admin'];?></td>
                                                <td class="center">
                                                <?php
```

```
            if($data['status']=='Dapur'){
            ?>
            <form name="pesan_dapur" method="post" action="updatepesanan.php">
                <input name="" type="submit" class="btn btn-success btn-block" value="Pesanan Selesai">
                <input type="hidden" name="notrans" value="<?php echo $data['notrans'];?>"/>
            </form>
            <?php
            }else{
            ?>
           <i class="btn btn-danger btn-block">Pesanan Sudah Selesai</i>
            <?php
            }
            ?>
            </td>
            <td class="center">
            <form name="pesan_dapur" method="post" action="cetakpesanan.php" target="_blank">
                <input name="" type="submit" class="btn btn-primary btn-block" value="Cetak">
                <input type="hidden" name="notrans" value="<?php echo $data['notrans'];?>"/>
            </form>
            </td>
        </tr>
    </tbody>
    <?php } ?>
</table>
```

C. Form Cetak Pesanan

```
<body>
<table width="400px" border="0" cellpadding="4" cellspacing="4"
style="border:1px solid #000">
 <?php
 include "koneksi.php";
 $notrans=$_POST['notrans'];
$tampil=mysqli_query($koneksi,"select * from transaksi where notrans= '$notrans' ");
$data=mysqli_fetch_array($tampil);
?>
  <tr>
    <td width="37%"><strong>No Transaksi</strong></td>
    <td width="63%"><?php echo $data['notrans'];?></td>
  </tr>
  <tr>
    <td><strong>No Meja</strong></td>
    <td><?php echo $data['no_mej'] ;?></td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td style="border-bottom:1px solid #000"><strong>Nama Menu</strong></td>
    <td style="border-bottom:1px solid #000"><strong>Jumlah</strong></td>
  </tr>
<?php
$no=1;
$tampil1=mysqli_query($koneksi,"select * from detail_transaksi where notrans='$notrans'");
while($data1=mysqli_fetch_array($tampil1)){
?>
  <tr>
    <td style="border-bottom:1px dashed #000"><?php echo $data1['nm_menu']?></td>

    <td style="border-bottom:1px dashed #000"><?php echo $data1['jmlh_pesanan']?></td>
  </tr>
<?php } ?>
</table>
</body>
</html>

<script>
    window.print() ;
</script>
```

4.4. Testing

A. Form Login Admin

Tabel IV.15

Hasil Pengujian Black Box Testing Form Login Admin

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan semua isian data login pada login restoran, lalu mengklik tombol “Sign in”	Username: (kosong) Password: (kosong)	Sistem akan menolak akses login “Username dan Password Tidak Valid”	Sesuai harapan	Valid
2	Hanya mengisi Username dan password kosong, lalu mengklik tombol “Sign in”	Username: (adm01) Password: (kosong)	Sistem akan menolak akses login “Username dan Password Tidak Valid”	Sesuai harapan	Valid
3	Username kosong dan password diisi, lalu mengklik tombol “Sign in”	Username: (kosong) Password: (12345678)	Sistem akan menolak akses login “Username dan Password Tidak Valid”	Sesuai harapan	Valid
4	Menginput dengan salah satu data yang benar saja benar, lalu klik tombol “Sign in”	Username: (adm01) Password: (1990-01-31)	Sistem akan menolak akses login “Username dan Password Tidak Valid”	Sesuai harapan	Valid
5	Menginput dengan data yang benar, lalu “Sign in”	Username: (adm01) Password: (12345678)	Sistem akan menerima dan bisa login untuk mengakses	Sesuai harapan	Valid

B. Form Login Pramusaji

Tabel IV.16

Hasil Pengujian Black Box Testing Form Login Pramusaji

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan semua isian data login pada login restoran, lalu mengklik tombol "Sign in"	User: (kosong) Password: (kosong)	Sistem akan menolak akses login "Username dan Password Tidak Valid"	Sesuai harapan	Valid
2	Hanya mengisi User dan password kosong, lalu mengklik tombol "Sign in"	User: (prs01) Password: (kosong)	Sistem akan menolak akses login "Username dan Password Tidak Valid"	Sesuai harapan	Valid
3	User kosong dan password diisi, lalu mengklik tombol "Sign in"	User: (kosong) Password: (12345678)	Sistem akan menolak akses login "Username dan Password Tidak Valid"	Sesuai harapan	Valid
4	Menginput dengan salah satu data yang benar saja benar, lalu klik tombol "Sign in"	User: (prs01) Password: (1990-01-31)	Sistem akan menolak akses login "Username dan Password Tidak Valid"	Sesuai harapan	Valid
5	Menginput dengan data yang benar, lalu "Sign in"	User: (prs01) Password: (12345678)	Sistem akan menerima dan bisa login untuk mengakses	Sesuai harapan	Valid

C. Form Konfirmasi Pemesanan

Tabel IV.17

Hasil Pengujian Black Box Testing Form Konfirmasi Pemesanan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan semua isian data pesanan pada konfirmasi pesanan, lalu mengklik tombol 'Konfirmasi'	No Meja: (kosong) Nama Pemesan: (kosong)	Sistem akan menolak dan memberikan pesan peringatan "Tidak Boleh Kosong!"	Sesuai harapan	Valid
2	Hanya mengisi No Meja dan Nama Pemesan kosong, lalu mengklik tombol 'Konfirmasi'	No Meja: (2) Nama Pemesan: (kosong)	Sistem akan menolak dan memberikan pesan peringatan "Tidak Boleh Kosong!"	Sesuai harapan	Valid
3	No Meja kosong dan Nama Pemesan diisi, lalu mengklik tombol 'Konfirmasi'	No Meja: (kosong) Nama Pemesan: (Siswara)	Sistem akan menolak dan memberikan pesan peringatan "Tidak Boleh Kosong!"	Sesuai harapan	Valid
4	No Meja diisi dengan karakter dan Nama Pemesan diisi, lalu mengklik tombol 'Konfirmasi'	No Meja: (satu) Nama Pemesan: (Siswara)	Sistem akan menolak dan memberikan pesan peringatan "Input Angka Max 2 Digit!"	Sesuai harapan	Valid
5	No Meja diisi dengan angka dan Nama Pemesan diisi dengan angka, lalu mengklik tombol 'Konfirmasi'	No Meja: (1) Nama Pemesan: (123)	Sistem akan menolak dan memberikan pesan peringatan "Input Karakter!"	Sesuai harapan	Valid
6	No Meja diisi dengan angka dan Nama Pemesan diisi dengan karakter, lalu mengklik tombol 'Konfirmasi'	No Meja: (1) Nama Pemesan: (Siswara)	Sistem akan menerima dan menyimpannya kedalam *database* dan tampil pesan "Data Sudah Tersimpan"	Sesuai harapan	Valid

D. Form Konfirmasi Pembayaran

Tabel IV.18

Hasil Pengujian Black Box Testing Form Konfirmasi Pembayaran

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan semua isian data pembayaran lalu mengklik tombol 'Konfirmasi'	Uang Bayar: (kosong) Uang Kembali: (kosong)	Sistem tidak akan merespons	Sesuai harapan	Valid
2	Mengisi Uang Bayar dengan nominal kurang dari total bayar, lalu mengklik tombol 'Konfirmasi'	Uang Bayar: (2000) (kurang)	Sistem akan menolak dan memberikan pesan peringatan "Pembayaran Kurang"	Sesuai harapan	Valid
3	Mengisi Uang Bayar sesuai dengan nominal dari total bayar, lalu mengklik tombol 'Konfirmasi'	Uang Bayar: (50000) (cukup)	Sistem akan menerima dan menyimpannya kedalam *database* dan tampil Struk Pembayaran	Sesuai harapan	Valid

4.5. Support

4.5.1. Publikasi Web

Mengakses *website* dengan menggunakan aplikasi *browser* seperti Mozilla Firefox, Internet Explorer, Opera, Google Chrome dan yang lainnya.Yaitu dengan mengetikan alamat *website* ke dalam *address bar* pada aplikasi *browser*.

Publikasi *website* Aplikasi Sistem Informasi Penjualan Restoran Caki Cake dilakukan dengan sistem intranet restoran yang memanfaatkan webhosting agar mempermudah petugas bila menjadi sebuah aplikasi. Selain itu dibutuhkan juga perangkat keras untuk menunjang kinerja aplikasi.

Untuk domain names, penulis mencoba mendaftarkan *website* Aplikasi Sistem Informasi pada http://www.idhostinger.com/ yaitu salah satu *web hosting* yang telah banyak dipergunakan oleh pembuat *web* untuk publikasikan.

Setelah mencoba mendaftar, telah didapatkan domain names sebagai berikut: http://www.cakicake.esy.es Adapun spesifikasi *hosting* yang penulis dapatkan pada penggunaan web hosting ini adalah:

a. *Disk Space* : 200 MB

b. *Bandwith* : 2 GB

c. *Email* : 2 Akun Email

d. *Subdomain* : *Unlimitted*

e. *Support* : PHP 5

f. MySql : *Unlimited*

g. PhpMyAdmin : *Supported*

4.5.2. Spesifikasi *Hardware* dan *Software*

Perangkat keras yang dimaksud disini adalah seperangkat alat atau elemen elektronik yang dapat membantu sistem yang diusulkan sehingga program yang diusulkan oleh penulis dapat bekerja dengan baik. Perangkat keras yang dibutuhkan dibagi atas dua bagian, yaitu perangkat keras untuk *web server* dan perangkat keras *client.* Sedangkan perangkat lunak adalah suatu rangkaian atau susunan instruksi yang harus benar dengan urutan-urutan yang benar pula.Keberadaan perangkat lunak selalu menyertai perangkat keras yang ada. Perangkat lunak yang dibutuhkan dibagi atas dua bagian, yaitu perangkat lunak untuk *web server* dan perangkat lunak untuk *client*.

1. Spesifikasi *Hardware*

Adapun perangkat keras minimal yang diperlukan oleh *server* adalah sebagai berikut:

a. *Processor* : *Intel Core 2 Duo 2.27 Ghz*

b. *Memory Size* (RAM) : 2,00 GB (1,87 GB *usable)*

c. *Monitor* : *SVGA Colour 15”*

d. *Harddisk* : 250 GB

e. *Keyboard* : 107 *Keys*

f. *Mouse* : *Standard Mouse*

Adapun perangkat keras minimal yang diperlukan oleh *client* adalah sebagai berikut:

a. *Processor* : *Intel Core 2 Duo 2.27 Ghz*

b. *Memory Size* (RAM) : 2,00 GB (1,87 GB *usable)*

c. *Monitor* : *LCD 14" LED*

d. *Harddisk* : 250 GB

e. *Keyboard* : 107 *Keys*

f. *Mouse* : *Standard Mouse*

g. *Printer* : *Deskjet*

2. Spesifikasi *Software*

Adapun perangkat lunak minimal yang diperlukan untuk *web server* adalah sebagai berikut:

a Sistem Operasi : *Windows seven*

b Bahasa Program : PHP

c *Interpreter* : *PHP5 versi 5.3.8*

d *Database Server* : *MySQL client version: mysqlnd 5.0.8-dev - 20102224 - $Revision: 310735 $*

e *Web Server* : *XAMPP versi 1.7.7 (for Windows)*

f *Database Tools* : *PhpMyAdmin versi 3.4.5*

Adapun perangkat lunak minimal yang diperlukan untuk *client* adalah sebagai berikut :

a Sistem Operasi : *Windows 9x/ME / 2000/ XP*

b Browser : *Mozzila Firefox 3.6*

4.6. Spesifikasi dokumen Sistem Usulan

1. Form Konfirmasi Pesanan

   Nama Dokumen : Konfirmasi Pesanan

   Fungsi : Sebagai *file* konfirmasi pesanan

   Sumber : Admin

   Tujuan : *Database*

   Frekuensi : Setiap ada Pesanan baru

   Media : *Monitor*

   Bentuk : Lampiran B.1

2. Data Menu

   Nama Dokumen : Daftar Menu

   Fungsi : Sebagai *file* Menu

   Sumber : Admin

   Tujuan : *Database*

Frekuensi : Setiap ada perubahan Menu

Media : *Monitor*

Bentuk : Lampiran B.2

3. Daftar Pesanan

Nama Dokumen : Daftar Pesanan

Fungsi : Sebagai *file* Pesanan

Sumber : Pelanggan

Tujuan : *Database*

Frekuensi : Setiap terjadi Pemesanan Menu

Media : *Monitor*

Bentuk : Lampiran B.3

4. Struk Penjualan

Nama Dokumen : Struk Penjualan

Fungsi : Sebagai *file* Penjualan

Sumber : Admin

Tujuan : *Database*

Frekuensi : Setiap ada Penjualan

Media : *Monitor*

Bentuk : Lampiran B.4

5. Data Laporan

Nama Dokumen : Data Laporan

Fungsi : Sebagai *file* Laporan

Sumber : Admin

Tujuan : *Database*

Frekuensi : Setiap ada Laporan

Media : *Monitor*

Bentuk : Lampiran B.5