**LAPORAN TUGAS BESAR**

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI**

**SISTEM INFORMASI KITCHEN RESTORAN SANGRILOKA**

**Tugas ini Diajukan untuk Memenuhi Tugas Besar Mata Kuliah   
Analisis dan Perancangan Sistem Informasi**



**Disusun Oleh :**

**Reynaldy Miftahudin Fauzi 5520115064**

**M.Aldy Rinaldi Djajadiredja 5520115072**

**M. Vikry Pratama 5520115067**

**Yadi Prassetya 5520115044**

**Nur’aini Siti Fatimah 5520115063**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS SURYAKANCANA**

**CIANJUR**

**2017/2018**

# LEMBAR *ASSESMENT*

**(SISTEM INFORMASI KITCHEN RESTORAN SANGRILOKA)**

***Team***

**(SIRELOKA)**

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK**

**UNIVERSITAS SURYAKANCANA**

**Laporan Tugas Besar APSI Ini Telah Disetujui dan Disahkan,**

|  |  |  |
| --- | --- | --- |
|  |  | **Cianjur, 31 Januari 2018** |
| ***Project Manager,***  **(Reynaldy Miftahudin Fauzi)**  **5520115064** |  | **Pembimbing,**  **(Regi Permana)**  **5520114053** |
| **Dosen Penanggung Jawab,**  **Siti Sarah Abdullah, M.T.**  **NIDN. 0403018001** | | |

# KATA PENGANTAR

Puji dan syukur kami panjatkan kepada Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada kami untuk dapat menyelesaikan laporan ini.Laporan yang berjudul “**Sistem Informasi Kitchen Restoran Sangriloka**” yang didalamnya mengandung beberapa materi seperti analisa, implementasi, perancangan sistem dan sebagainya.

Dengan segala keterbatasan pengetahuan serta pengalaman kami dalam penyusunan laporan ini, kami menyadari bahwa masih banyak kekurangannya, baik dalam hal perancangan aplikasi maupun sistematika penulisannya. Namun berkat bimbingan dari berbagai pihak yang dengan sabar mengarahkan kami, sehingga akhirnya kami dapat menyelesaikan laporan ini.

Pada kesempatan ini pula kami menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam penyusunan laporan ini, terima kasih terutama kami ucapkan kepada :

1. Ibu Siti Sarah Abdullah, M.T. selaku dosen mata kuliah Sistem Informasi yang telah mendukung dan membimbing kami dalam melaksanakan penelitian untuk tugas Sistem Informasi ini.
2. Regi Permana selaku asisten pembimbing Analisis dan Perancangan Sistem Informasi yang telah memberikan waktu dan ilmunya sehingga kami dapat menerapkannya kedalam proses analisis dan perancangan sistem informasi ini.
3. Beserta seluruh rekan-rekan seperjuangan dan semua pihak yang terlibat dalam pembuatan laporan ini yang tidak bisa kami sebutkan satu persatu.

Semoga Allah SWT membalas segala bantuan yang diberikan semua pihak dalam menyelesaikan penulisan laporan Sistem Informasi ini, Amin.

Cianjur, 24 Januari 2017

Penulis,

**DAFTAR ISI**

[LEMBAR *ASSESMENT* i](#_Toc506232138)

[KATA PENGANTAR ii](#_Toc506232139)

[DAFTAR TABEL v](#_Toc506232140)

[DAFTAR GAMBAR vi](#_Toc506232141)

[DAFTAR SIMBOL viii](#_Toc506232142)

[DAFTAR ISTILAH xvii](#_Toc506232143)

[BAB I PENDAHULUAN 18](#_Toc506232144)

[1.1. Judul Analisis dan Perancangan Sistem Infomasi 18](#_Toc506232145)

[1.2. Definisi Analisis dan Perancangan Sistem Informasi 18](#_Toc506232146)

[1.3. Lingkup Sistem 18](#_Toc506232147)

[1.4. Batasan Sistem 19](#_Toc506232148)

[1.5. Waktu Pelaksanaan 20](#_Toc506232149)

[Tabel 1.1 Penjadwalan 20](#_Toc506232150)

[BAB II TINJAUAN PUSTAKA 21](#_Toc506232151)

[2.1. Landasan Teori 21](#_Toc506232152)

[2.1.1. Definisi Analisis 21](#_Toc506232156)

[2.1.2. Definisi Perancangan 22](#_Toc506232161)

[2.1.3. Definisi Sistem 22](#_Toc506232162)

[2.1.4. Definisi Informasi 22](#_Toc506232163)

[2.1.5. Definisi Sistem Informasi 22](#_Toc506232164)

[2.1.6. Definisi *Unified Modelling Language* (UML) 23](#_Toc506232165)

[2.2. *Tool-Tool* Yang Digunakan 25](#_Toc506232166)

[2.2.1. Visual Paradigm 25](#_Toc506232171)

[2.2.2. Sublime Text 3 26](#_Toc506232172)

[2.2.3. GitHub 26](#_Toc506232173)

[2.2.4. Balsamiq Mockups 3 27](#_Toc506232174)

[2.2.5. Mozilla Firefox 27](#_Toc506232175)

[2.2.6. XAMPP 27](#_Toc506232176)

[BAB III *ANALYSIS MODELING* 28](#_Toc506232177)

[3.1. *Scenario-Based Modeling* 28](#_Toc506232178)

[3.1.1. *User Scenario* 28](#_Toc506232183)

[Tabel 3.1 *User Scenario* 28](#_Toc506232184)

[3.1.2. *Actor List* 32](#_Toc506232185)

[Tabel 3.2 *Actor List* 32](#_Toc506232186)

[3.1.3. *Use Case List* 32](#_Toc506232187)

[Tabel 3.3 *Use Case List* 32](#_Toc506232188)

[3.1.4. Use Case Diagram 33](#_Toc506232189)

[3.1.5. Swimlane Diagram 34](#_Toc506232190)

[a. Swimlane Alat Makan 34](#_Toc506232191)

[b. Swimlane Kelola Order 35](#_Toc506232192)

[c. Swimlane Kelola Stok 36](#_Toc506232193)

[d. Swimlane Pengajuan Stok 37](#_Toc506232194)

[e. Swimlane Penerimaan Stok 38](#_Toc506232195)

[3.2. *Data Modelling* 39](#_Toc506232196)

[3.2.1. *Class Diagram* 39](#_Toc506232197)

[3.2.2. *Entity Relationship Diagram* 40](#_Toc506232198)

[3.3. *Behavioral Modelling* 40](#_Toc506232199)

[3.3.1. *State Diagram* 40](#_Toc506232200)

[a. State Diagram Alat Makan 41](#_Toc506232201)

[b. State Diagram Kelola Order 42](#_Toc506232202)

[c. State Diagram Kelola Stok 42](#_Toc506232203)

[d. State Diagram Pengajuan Stok 43](#_Toc506232204)

[e. State Diagram Penerimaan Stok 43](#_Toc506232205)

[3.3.2. *Sequence Diagram* 44](#_Toc506232206)

[a. Sequence Diagram Alat Makan 44](#_Toc506232207)

[b. Sequence Diagram Kelola Order 45](#_Toc506232208)

[c. Sequence Diagram Kelola Stok 46](#_Toc506232209)

[d. Sequence Diagram Pengajuan Stok 47](#_Toc506232210)

[e. Sequence Diagram Penerimaan Stok 47](#_Toc506232211)

[BAB IV *DESIGN MODELLING* 48](#_Toc506232212)

[4.1. *Architectural Design* 48](#_Toc506232217)

[4.1.1. Struktur Menu 48](#_Toc506232218)

[4.2. *User Interface Design* 49](#_Toc506232219)

[4.3. *Component Level Design* 53](#_Toc506232220)

[4.3.1. Algoritma 53](#_Toc506232221)

[4.4. *Deployment Level Design* 59](#_Toc506232222)

[BAB V PENUTUP 60](#_Toc506232223)

[5.1. Kesimpulan 60](#_Toc506232224)

[5.2. Saran 60](#_Toc506232225)

[DAFTAR PUSTAKA 61](#_Toc506232226)

[LAMPIRAN 62](#_Toc506232227)

# DAFTAR TABEL

Tabel 1.1 Penjadwalan 21

Tabel 3.1 *User Scenario* 30

Tabel 3.2 *Actor List* 33

Tabel 3.3 *Use Case List* 33

# DAFTAR GAMBAR

[Gambar 3.1 Use Case Diagram 33](#_Toc504833405)

[Gambar 3.2 *Swimlane Diagram* Alat Makan 34](#_Toc504833406)

[Gambar 3.3 *Swimlane Diagram* Kelola Order 35](#_Toc504833407)

[Gambar 3.4 *Swimlane Diagram* Kelola Stok 36](#_Toc504833408)

[Gambar 3.5 *Swimlane Diagram* Pengajuan Stok 37](#_Toc504833409)

[Gambar 3.6 Swimlane Diagram Penerimaan Stok 38](#_Toc504833410)

[Gambar 3.7 *Class Diagram* SIRELOKA 39](#_Toc504833411)

[Gambar 3.8 ERD SANGRILOKA 40](#_Toc504833412)

[Gambar 3.9 *State Diagram* Alat Makan 41](#_Toc504833413)

[Gambar 3.10 *State Diagram* Kelola Order 42](#_Toc504833414)

[Gambar 3.11 *State Diagram* Kelola Stok 42](#_Toc504833415)

[Gambar 3.12 *State Diagram* Pengajuan Stok 43](#_Toc504833416)

[Gambar 3.13 *State Diagram* Penerimaan Stok 43](#_Toc504833417)

[Gambar 3.14 *Sequence Diagram* Alat Makan 44](#_Toc504833418)

[Gambar 3.15 *Sequence Diagram* Kelola Order 45](#_Toc504833419)

[Gambar 3.16 *Sequence Diagram* Kelola Stok 46](#_Toc504833420)

[Gambar 3.17 *Sequence Diagram* Pengajuan Stok 47](#_Toc504833421)

[Gambar 3.18 *Sequence Diagram* Penerimaan Stok 47](#_Toc504833422)

[Gambar 4.1 Struktur Menu SIRELOKA........................................................................... 51](#_Toc504833643)

[Gambar 4.2 *User Interface Design* Halaman Kelola Order 52](#_Toc504833644)

[Gambar 4.3 *User Interface Design* Halaman Stok Bahan Makan 53](#_Toc504833645)

[Gambar 4.4 *User Interface Design* Halaman Peralatan Makan 53](#_Toc504833646)

[Gambar 4.5 *User Interface Design* Halaman Pengajuan Stok 54](#_Toc504833647)

[Gambar 4.6 *User Interface Design* Halaman Penerimaan Stok 55](#_Toc504833648)

# DAFTAR SIMBOL

1. **Simbol *Use Case Diagram***

| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| --- | --- | --- | --- |
| 1 |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *Use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *Use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *Use case* target memperluas perilaku dari *Use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

1. **Simbol *Swimlane Diagram***

| **No** | **Nama Simbol** | **Gambar** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | *Start State* |  | Menggambarkan awal dari suatu aktivitas yang berjalan pada sistem |
| 2 | *End State* |  | Menggambarkan akhir dari suatu aktivitas yang berjalan pada sistem. |
| 3 | *Activity* |  | Menggambarkan aktivitas yang dilakukan pada sistem. |
| 4 | *Swimlane* |  | Menggambarkan pembagian atau pengelompokan berdasarkan tugas dan fungsi tersendiri. |
| 5 | *Transition to self* |  | Menggambarkan hubungan antara *state* atau *activity* yang kembali kepada *state* atau *activity* itu sendiri. |
| 6 | *Transition State* |  | Menggambarkan hubungan antara dua *state*, dua *activity* ataupun antara *state* dan *activity.* |
| 7 | *Decision* |  | Menggambarkan kondisi dari sebuah aktivitas yang bernilai benar atau salah. |
| 8 | *State* |  | Menggambarkan kondisi, situasi ataupun tempat untuk beberapa aktivitas. |
| 9 | *Fork* |  | Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan. |
| 10 | *Join* |  | Menggambarkan aktivitas yang dimulai dengan dua atau lebih aktivitas yang sudah dilakukan dan menghasilkan sebuah aktivitas. |

1. **Simbol *Class Diagram***

| **No** | **Nama Simbol** | **Gambar** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | *Class Diagram* |  | *Class* menggambarkan keadaan (atribut/properti) dari suatu objek. Nama menggambarkan nama dari *Class*/objek. Atribut menggambarkan batasan dari nilai yang dapat dimiliki oleh *property* tersebut. *Method* menggambarkan implementasi dari layanan yang dapat diminta dari beberapa *object* dari *Class* , yang mempengaruhi *behaviour.* |
| 2 | *Association* |  | Menggambarkan mekanisme komunikasi suatu objek dengan objek lainnya. Atau dapat juga menggambarkan ketergantungan antar kelas. |
| 3 | *Aggregate* |  | Menggambarkan bahwa suatu objek secara fisik dibentuk dari objek-objek lain. |
| 4 | *Multiplicity* |  | Menggambarkan banyaknya object yang terhubung satu dengan yang lainnya. Contoh :  1 Tepat satu  0..\* Nol atau lebih  1..\* Satu atau lebih  0..1 Nol atau satu |

1. **Simbol *Entity Relationship Diagram***

| **No.** | **Nama Simbol** | **Gambar** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | *Entity* |  | Menggambarkan himpunan entitas ini bisa berupa: suatu elemen lingkungan, sumber daya, atau transaksi yang ada dalam lingkungan perusahaan. |
| 2 | *Attribute* |  | Menggambarkan karakter atau nama-nama atribut yang ada pada *Entity.* |
| 3 | *Primary Key Attribute* |  | Menggambarkan atribute yang berfungsi sebagai Key (kunci) yang ada pada suatu *Entity*. |
| 4 | *Relationship* |  | Menggambarkan relasi yang digunakan untuk menunjukkan hubungan yang ada antara *Entity* satu dengan *Entity* lainnya. |
| 5 | *Garis/Link* |  | Menggambarkan garis penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan Atributnya. |

1. **Simbol *Sequence Diagram***

| **No** | **Nama Simbol** | **Gambar** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | Aktor |  | Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. |
| 2 | *Boundary* |  | Menggambarkan interaksi antara satu atau lebih *actor* dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar. |
| 3 | *Control* |  | Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem. |
| 4 | *Entity* |  | Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem). |
| 5 | *Object Message* |  | Menggambarkan pesan/hubungan antar objek, yang menunjukan urutan kejadian yang terjadi. |
| 6 | *Message to Self* |  | Menggambarkan pesan/hubungan objek itu sendiri, yang menunjukan urutan kejadian yang terjadi. |
| 7 | *Object* |  | Menggambarkan abstraksi dari sebuah entitas nyata atau tidak nyata yang informasinya harus disimpan. |

1. **Simbol *State Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *State* | Nilai atribut dan nilai *link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah *state* objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

# DAFTAR ISTILAH

|  |  |  |
| --- | --- | --- |
| **No.** | **Istilah** | **Keterangan** |
| 1 | SIRELOKA | Sistem Informasi Kitchen Restoran Sangriloka |
| 2 | ERD | *Entity Relationship* Diagram adalah merupakan salah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah *database* |
| 3 | UML | *Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta – model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). |

# BAB I PENDAHULUAN

Bab ini membahas tentang judul analisis dan perancangan sistem informasi, definisi dari analisis dan perancangan sistem informasi, lingkup dan batasan sistem, dan waktu pelaksanaan kegiatan.

## Judul Analisis dan Perancangan Sistem Infomasi

Sistem Informasi Kitchen Restoran Sarilongka

## Definisi Analisis dan Perancangan Sistem Informasi

Menganalisa proses penyediaan stok untuk menentukan apa masalah yang ada pada sistem administrasi disana dan merancang Sistem Informasi untuk mengatasi permasalahan stok yang muncul maupun mungkin muncul di bagian dapur.

## Lingkup Sistem

Lingkup sistem yang dimaksud adalah sebagai berikut :

* 1. Alat Makan

Meliputi perlengkapan makan yang disediakan restoran untuk konsumen, berupa sendok, garpu, piring, gelas dan tisu.

* 1. Bahan Makanan

Bahan makanan adalah bahan yang bisa dijadikan makanan. Bahan makanan berperan penting dalam usaha dan sistem dalam restoran

* 1. Kelola Order

Kelola Order merupakan proses untuk mengelola pembelian suatu bahan makanan dan alat makan kepada si penjual. Tujuannya agar barang yang datang sesuai dengan apa yang dibutuhkan di dapur.

* 1. Kelola Stok

Suatu tatanan untuk mengelola pendistribusian barang-barang agar barang yang tersimpan tetap dalam keadaan baik, kualitas bahan makanan bersih dan didistribusikan kepada para koki dalam jumlah yang tepat.

* 1. Pengajuan Stok

Proses pengiriman surat pengajuan oleh seorang admin di bagian dapur yang ditujukan kepada penjual untuk dapat mengirimkan alat makan maupun bahan makanan yang diperlukan.

* 1. Penerimaan Stok

Dalam kasus ini, admin menerima barang berupa stok bahan dan alat dapur dari pengajuan stok. Barang diterima serta dicek apakah sesuai atau tidak.

## Batasan Sistem

Batasan sistem yang dimaksud adalah sebagai berikut :

1. Transaksi

Transaksi merupakan suatu kegiatan yang diakukan seseorang yang menimbulkan perubahan terhadap harta atau keuangan yang dimiliki baik itu bertambah ataupun berkurang. Misalnya menjual harta, membeli barang, membayar hutang, serta membayar berbagai macam biaya untuk memenuhi kebutuhan hidup.

1. Kelola Menu

Merupakan suatu kegiatan mengatur daftar menu makanan yang akan disediakan pada pelanggan, dari daftar harga makanan kelas biasa sampai kelas mewah(VIP).

1. Penggajian Pegawai

Penggajian di sini meliputi proses pengupahan terhadap pegawai. Sistem ini mempunyai deadline dengan menentukan tanggal penggajian. Di mana menilai potensi dari kinerja pegawai.

## Waktu Pelaksanaan

Waktu yang diperlukan dalam pembuatan laporan Analisis dan Perancangan Sistem Informasi Kitchen Restoran Sarilongka sebagai berikut :

### Tabel 1.1 Penjadwalan

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kegiatan | Waktu Pelaksanaan | | | | | | | | |
| November | | | Desember | | | | Januari | |
| 04 | 17 | 24 | 05 | 12 | 16 | 25 | 5 | 21 |
| Penggambaran sistem yang dibuat |  |  |  |  |  |  |  |  |  |
| Pembuatan *scenario based modeling* |  |  |  |  |  |  |  |  |  |
| Pembuatan *use case* |  |  |  |  |  |  |  |  |  |
| Pembuatan *swimlane* |  |  |  |  |  |  |  |  |  |
| Pembuatan *class* *Diagram* |  |  |  |  |  |  |  |  |  |
| Pembuatan *state* *Diagram* |  |  |  |  |  |  |  |  |  |
| Pembuatan *intarface* |  |  |  |  |  |  |  |  |  |
| Pembuatan aplikasi |  |  |  |  |  |  |  |  |  |
| Penyusunan Laporan |  |  |  |  |  |  |  |  |  |

# BAB II TINJAUAN PUSTAKA

Di dalam bab ini membahas mengenai landasan teori yang digunakan sebagai acuan dalam membuat laporan, definisi perancangan dan definisi-definisi lainnya, serta *tool-tool* yang digunakan untuk merancang Sistem Informasi Kitchen Restoran Sangriloka.

## Landasan Teori

Landasan teori merupakan teori yang relevan yang digunakan untuk menjelaskan variable yang akan diteliti dan sebagai dasar untuk memberi jawaban sementara terhadap rumusan masalah yang diajukan (hipotesis), dan penyusunan instrument penelitian. Teori yang digunakan bukan sekedar pendapat dari pengarang atau pendapat lain, tetapi teori yang benar-benar telah diuji kebenarannya.



### Definisi Analisis

Analisa berasal dari kata Yunani Kuno “analusis” yang berarti melepaskan. Analusis terbentuk dari dua suku kata yaitu “ana” yang berarti kembali dan “luein” yang berarti melepas. Sehingga **pengertian analisis** yaitu suatu usaha dalam mengamati secara detail pada suatu hal atau benda dengan cara menguraikan komponen-komponen pembentuknya atau menyusun komponen tersebut untuk dikaji lebih lanjut.

Dalam Kamus Bahasa Indonesia Kontemporer karangan Peter Salim dan Yenni Salim (2002) menjabarkan pengertian analisis sebagai berikut :

Analisis adalah penyelidikan terhadap suatu peristiwa (perbuatan, karangan dan sebagainya) untuk mendapatkan fakta yang tepat (asal usul, sebab, penyebab sebenarnya, dan sebagainya).



### Definisi Perancangan

Perancangan adalah suatu kegiatan yang memiliki tujuan untuk men*design* sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. (Al-Bahra Bin Ladjamudin. 2005:39).

### Definisi Sistem

Sistem adalah kumpulan elemen yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan. Elemen tersebut terdiri dari elemen konvensional (data, manusia, prosedur) dan elemen modern (data, manusia, prosedur, *hardware* , *software*). (Yulianto, Ardhian Agung, dkk. 2009:2).Suatu sistem menerima masukan, memproses, dan menghasilkan keluaran. Sistem dapat bekerja karena komponen didalamnya saling berinteraksi untuk menghasilkan keluaran. Dalam prosesnya, kinerja sistem dipengaruhi oleh lingkungan disekitarnya.

### Definisi Informasi

Menurut Gordon (1992) informasi merupakan sekumpulan data / fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima.

### Definisi Sistem Informasi

Menurut Ladjamudin (2005) Sistem Informasi merupakan suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang tergorganisasi.

### Definisi *Unified Modelling Language* (UML)

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta – model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar – standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar – standar CORBA (*Common Object Request Broker Architecture*) (Fowler Martin, 2005).

Menurut Verdi Yasin (2012:194) UML adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. seperti bahasa - bahasa lainnya, UML mendefinisikan notasi dan *syntax/* semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai *Diagram* piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: *OOD (Object-Oriented Design), OMT (Object Modeling Technique), dan OOSE (Object-Oriented Software Engineering.*

Menurut Verdi Yasin (2012:194) di dalam UML terdapat beberapa *Diagram* yaitu:

*Use case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem.

*Class Diagram*

*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

*Statechart Diagram*

*Statechart Diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima.

*Activity Diagram*

*Activity Diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

*Sequence Diagram*

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

*Collaboration Diagram*

*Collaboration Diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing – masing objek dan bukan pada waktu penyampaian *message.*

*Component Diagram*

*Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya.

*Deployment Diagram*

*Deployment Diagram* menggambarkan detil bagaimana komponen *di-deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal – hal lain yang bersifat fisikal.

## *Tool-Tool* Yang Digunakan

Agar memudahkan dalam pembuatan sistem, maka diperlukan beberapa alat atau *tools-tools* yang digunakan dalam pembuatan Sistem Informasi Kitchen Restoran Sangriloka.

*Tools* – *tools* yang digunakan memaparkan perangkat apa saja yang digunakan dalam membangun aplikasi Sistem Informasi Kitchen Restoran Sangriloka. Untuk membangun sebuah aplikasi berbasis *website*, diperlukan beberapa perangkat seperti bahasa pemrograman,*tools* perancangan sistem,*editor,* dan *database*. Berikut ini adalah bahasa pemrograman, *editor*, dan *database* yang digunakan dalam membangun aplikasi.



### Visual Paradigm

*Visual Paradigm* merupakan sebuah *software* model dengan sistem visualisasi yang memungkinkan model yang telah dibuat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur yang ada didalamnya sampai pada menganalisa sebuah proyek yang akan dikerjakan. *Diagram* dapat disusun sedemikian rupa sehingga dapat dipustakakan menjadi proyek per proyek yang saling berkaitan. Hal ini dapat juga membantu memisahkan terhadap pekerjaan proyek sampai *level* terkecil.

*Visual Paradigm* dapat membandingkan perubahan antara *diagram* yang satu dengan yang lain yang saling berkaitan dan dapat digunakan untuk melacak perbedaan antara *Diagram*-digran yang ada. Dengan dukungan *design* layaknya *freehand* dapat menambahkan ke *diagram* untuk penjelasan lebih detail apa yang menjadi tujuan. Fasilitas lain dapat menggunakan mouse pen atau pena untuk sketching bentuk jalan, kita juga dapat menambahkan bentuk solid seperti bintang, dan panah *rectangles*.

### Sublime Text 3

Sublime text merupakan salah satu text editor yang dapat digunakan oleh programmer, khususnya web developer. Sublime text memiliki beberapa kelebihan dari text editor lainnya, diantaranya :

1. *Multiple Selection*

Mempunyai fungsi untuk melakukan perubahan pada sebuah kode dalam waktu yang sama dan  dalam baris yang berbeda.

1. *Command Pallete*

Mempunyai fungsi yang berguna untuk mengakses file shortcut dengan mudah,untuk mencari file  tersebut dengan menekan CTRL+SHIFT+P

1. *Distraction Free Mode*

Fitur ini sangat dibutuhkan oleh pengguna yang sedang fokus dalam pekerjaan,yaitu dapat merubah  tampilan layar menjadi penuh dengan menekan SHIFT+F11

1. *Find In Project*

Memudahkan pengguna dalam memilih dan mencari file dalam project dengan menekan SHIFT+P.

### GitHub

GitHub adalah [layanan penginangan web](https://id.wikipedia.org/wiki/Layanan_penginangan_web) bersama untuk proyek pengembangan [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) yang menggunakan sistem [pengontrol versi](https://id.wikipedia.org/wiki/Pengontrol_versi) [Git](https://id.wikipedia.org/wiki/Git) dan [layanan hosting internet](https://id.wikipedia.org/wiki/Layanan_hosting_internet).

### Balsamiq Mockups 3

*Mockup* artinya model atau replika mesin atau struktur, yang digunakan untuk tujuan instruksional atau eksperimental.

*Balsamiq mockup* adalah program aplikasi yang digunakan dalam pembuatan tampilan *user interface* sebuah aplikasi. *Software* ini sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain prototyping aplikasi yang akan kita buat. *Software* ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

### Mozilla Firefox

Mozilla Firefox digunakan sebagai alat sekaligus tempat untuk menjalankan aplikasi Sistem Informasi Kitchen Restoran Sangriloka.

### XAMPP

XAMPP adalah perangkat lunak yang menghubungkan tiga aplikasi kedalam satu paket yaitu Apache, MySQL, dan PHPMyAdmin. XAMPP ini merupakan sebuah perangkat lunak yang mudah digunakan dan mendukung instalasi di Linux dan Windows.

# BAB III *ANALYSIS MODELING*

*Analysis Modeling* membahas tentang alur dari perancangan sebuah sistem yang akan dibangun meliputi *Scenario-Based Modeling, Data Modeling dan Behavioral Modeling.*

## *Scenario-Based Modeling*

*Scenario-Based Modeling* menjelaskan bagaimana memahami interaksi antara *user* dengan sistem. Didalam *scenario-based modeling* produk yang dihasilkan adalah *Use case* diagram dan *activity* diagram atau *Swimlane Diagram*. (Pressman, 2010).

Model ini merupakan model yang bergantung pada permasalahan yang sekuensial dan terfokus pada proses. Elemen model ini meliputi *Use cases* – *text , Use case Diagrams*, *Activity Diagrams* dan *Swim lane Diagrams*.

Karakteristik pemodelan ini memiliki permasalahan terbagi-bagi menjadi tahapan-tahapan proses. Tahapan dan proses selanjutnya dapat dianalisis menjadi *use case* yang sesuai, saling berkaitan antara proses yang satu dan proses lainnya. Interaksi terhadap *user* menentukan bagaimana proses akan terbentuk.



### *User Scenario*

*User Scenario* adalah rangkaian kalimat yang bersifat narasi atau bersifat menguraikan (menjelaskan dsb, dalam makna lain *user scenario* di katakan sebagai prosa yang subjeknya merupakan suatu rangkaian kejadian). Berikut adalah *User Scenario* yang ada pada Sistem Informasi Kitchen Restoran Sangriloka.

### Tabel 3.1 *User Scenario*

| **Kode** | ***Use Case*** | **Aktor** | ***Activity*** |
| --- | --- | --- | --- |
| U1 | Alat Makan | Admin | * Admin memilih menu alat makan. * Kemudian Admin memilih tombol aksi sesuai dengan kehendak. * Admin memilih tombol tambah alat makan. * Admin mengisi form alat makan. * Lalu admin menekan tombol simpan. * Admin memilih tombol cari alat makan. * Admin mengisi form cari alat makan. * Lalu admin menekan tombol cari. * Admin memilih tombol edit alat makan. * Admin mengisi form edit alat makan. * Lalu admin menekan tombol edit. * Admin memilih tombol hapus alat makan. * Lalu admin memilih tombol hapus. |
| U2 | Kelola Order | Admin | * Admin memilih menu kelola order. * Admin melihat orderan. * Admin menekan tombol status order. * Admin memilih tombol pesanan habis. * Admin memilih tombol sedang dimasak. * Admin memilih tombol selesai dimasak. |
| U3 | Kelola Stok | Admin | * Admin memilih menu kelola stok. * Kemudian admin menekan tombol cek makanan. * Admin melihat makanan tidak layak. * Lalu admin menekan tombol kurangi stok makanan. |
| U4 | Pengajuan Stok | Admin | * Admin memilih menu pengajuan stok. * Admin mengisi form stok yang akan diajukan. * Lalu admin menekan tombol kirim. |
| U5 | Penerimaan Stok | Admin | * Admin memilih menu penerimaan stok. * Admin melihat barang yang telah dikirim. * Lalu admin menekan tombol terima barang. |

### *Actor List*

Aktor merupakan sejumlah orang (atau sarana) yang berbeda yang menggunakan sistem atau produk di dalam konteks fungsi-fungsi dan perilaku-perilaku yang harus dideskripsikan selanjutnya. Aktor adalah segala sesuatu yang berkomunikasi dengan sistem/produk dan pada dasarnya bersifat eksternal untuk sistem/perangkat itu sendiri.

Identifikasi aktor ini menjelaskan tentang siapa saja aktor atau *user* yang terlibat dengan Sistem Informasi Kitchen Restoran Sangriloka.

### Tabel 3.2 *Actor List*

|  |  |
| --- | --- |
| **Aktor** | **Keterangan** |
| Admin | Orang yang mengelola semua data pada Sistem Informasi Kitchen Restoran Sangriloka. |

### *Use Case List*

Daftar *Use case* menjelaskan tentang daftar *use case* yang ada didalam sistem, daftar *use case* berisi kode, nama *use case* dan keterangan dari tiap masing-masing *use case*.

### Tabel 3.3 *Use Case List*

|  |  |  |
| --- | --- | --- |
| Kode | Sistem *UseCase* | Keterangan |
| U1 | Alat Makan | Admin dapat tambah, edit, cari, hapus data alat makan. |
| U2 | Kelola Order | Admin mengupdate status dari orderan. |
| U3 | Kelola Stok | Admin dapat mengecek stok bahan makanan, dan mengurangi stok bahan makanan. |
| U4 | Pengajuan Stok | Admin dapat mengajukan stok dan mengisi form pengajuan stok. |
| U5 | Penerimaan Stok | Admin mengecek barang yang telah dikirim dan menerima barang. |

### Use Case Diagram

*Use case diagram* mendeskripsikan kelakuan sistem dari sudut pandang pengguna, berguna untuk membantu memahami kebutuhan. *Use case* adalah dasar dari diagram lain. *Use case* adalah abstraksi dari interaksi antara *system* dan *actor*. *Use case*  berkerja dengan mendeskripsikan tipe interaksi antara *actor* sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai.

|  |
| --- |
| **Kelola** |
| Gambar 3.1 Use Case Diagram |

Keterangan :

Gambar *use case* pengelolaan *kitchen* tersebut memilik lima fungsi dan satu aktor. Untuk fungsi terdiri dari U1.Alat Makan, U2.Kelola Order, U3.Stok Bahan Makanan, U4.Pengajuan Stok, dan U5.Penerimaan Stok. Sedangkan aktor hanya admin saja.

### Swimlane Diagram

*Swimlane Diagram* merupakan diagram yang memperlihatkan aliran aktivitas-aktivitas yang dideskripsikan oleh *use case* dan pada saat yang bersamaan memperlihatkan aktor mana atau kelas analisa mana yang bertanggungjawab untuk aksi tertentu yang dideskripsikan oleh kotak aktivitas.

#### Swimlane Alat Makan

|  |
| --- |
|  |
| Gambar 3.2 *Swimlane Diagram* Alat Makan |

Keterangan :

Gambar *Swimlane Diagram* Alat makan menunjukan aktifitas Admin melakukan aksi/tindakan untuk mengubah data pada alat makan sesuai yang dibutuhkan. Admin bisa memilih tambah alat makan, cari alat makan, edit alat makan, dan hapus alat makan. Jika terjadi kesalahan pada saat mengisi form tambah dan edit maka sistem akan kembali ke tampilan mengisi form, dan untuk cari jika data tidak ditemukan maka akan muncul pesan data tidak ditemukan. Setelah memilih aksi/tindakan admin akan menekan tombol simpan ataupun cari. Dan akan kembali pada halaman awal alat makan.

#### Swimlane Kelola Order

|  |
| --- |
|  |
| Gambar 3.3 *Swimlane Diagram* Kelola Order |

Keterangan :

Gambar *Swimlane Diagram* Kelola Order menunjukan aktifitas admin saat akan melihat list orderan. Pada saat melihat *list* orderan, admin menekan tombol status order yang bertujuan untuk meng-update data order. Terdapat tiga status order yaitu pesanan habis, pesanan sedang dimasak, dan pesanan selesai dimasak.

#### Swimlane Kelola Stok

|  |
| --- |
|  |
| Gambar 3.4 *Swimlane Diagram* Kelola Stok |

Keterangan :

Gambar *Swimlane Diagram* Kelola Stok menunjukan aktifitas Admin saat mengakses halaman kelola stok. *Interface* akan menampilkan tidak ada bahan makanan yang tidak layak dan menampilkan alert untuk bahan makanan yang sudah tidak layak. Admin menekan tombol “cek makanan” untuk melihat bahan makanan yang tidak layak pakai, lalu menekan tombol kurangi stok makanan. Setelah itu interface menampilkan pesan stok makanan berhasil dikurangi.

#### Swimlane Pengajuan Stok

|  |
| --- |
|  |
| Gambar 3.5 *Swimlane Diagram* Pengajuan Stok |

Keterangan :

Gambar *Swimlane Diagram* Pengajuan Stok menunjukan aktifitas admin saat mengakses menu pengajuan stok. Admin akan mengisi form untuk stok apa yang akan diajukan, setelah mengisi form, admin menekan tombol kirim, lalu *interface* menampilkan pesan pengajuan stok terkirim.

#### Swimlane Penerimaan Stok

|  |
| --- |
|  |
| Gambar 3.6 Swimlane Diagram Penerimaan Stok |

Keterangan :

Gambar *Swimlane Diagram* Penerimaan Stok menunjukan aktifitas admin saat mengakses menu penerimaan stok. Sistem akan menampilkan barang yang telah dikirim, lalu admin akan menekan tombol terima barang. Setelah itu *interface* menampilkan pesan barang diterima.

## *Data Modelling*

*Data Modeling* adalah object data yang diproses didalam sistem beserta relasi antar *object* data. *Object* data merupakan representasi dari gabungan informasi yang harus dipahami oleh *software* atau sistem. (Pressman, 2010).

### *Class Diagram*

*Class Diagram* adalah pandangan aplikasi yang statis. *Class* *Diagram* tidak hanya menggambarkan visualisasi, menggambarkan dan mendokumentasikan aspek yang berbeda dalam sistem, tetapi juga untuk kontruksi eksekusi kode dalam *software* aplikasi.

*Class diagram* digunakan untuk mengelompokkan hal-hal inti dari setiap proses yang ingin dilakukan. Semua proses dimasukkan ke dalam tiap-tiap *class* dan saling dihubungkan pada *class*-*class* lainnya yang saling berhubungan.

|  |
| --- |
|  |
| Gambar 3.7 *Class Diagram* SIRELOKA |

### *Entity Relationship Diagram*

*Entity Relationship* Diagram adalah merupakan salah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah *database.* Umumnya setelah perancangan ERD selesai berikutnya adalah mendesain *database* secara fisik yaitu pembuatan tabel, index dengan tetap mempertimbangkan *performance*. Kemudian setelah *database* selesai dilanjutkan dengan merancang aplikasi yang melibatkan *database.*

|  |
| --- |
|  |
| Gambar 3.8 ERD SANGRILOKA |

## *Behavioral Modelling*

*Behavioral Modeling* menunjukan bagaimana sebuah *software* akan merespon terhadap kejadian diluar atau didalam sistem. (Pressman, 2010).

### *State Diagram*

*State Diagram* adalah suatu metode untuk merepresentasikan jalannya suatu sistem yang digambarkan berdasarkan kejadian sebab sistem tersebut keadaanya harus diubah. *State Diagram* menunjukan sebuah aksi atau proses aktifasi berdasarkan akibat dari kejadian yang berupa fakta.

*Diagram* *state* UML merepresentasikan keadaan-keadaan (*state*) aktif untuk masing-masing kelas dan *event-event* (pemicu) yang menyebabkan perubahan-perubahan diantara keadaan-keadaan (*state)*  aktif tersebut.

#### State Diagram Alat Makan

|  |
| --- |
|  |
| Gambar 3.9 *State Diagram* Alat Makan |

#### State Diagram Kelola Order

|  |
| --- |
|  |
| Gambar 3.10 *State Diagram* Kelola Order |

#### State Diagram Kelola Stok

|  |
| --- |
|  |
| Gambar 3.11 *State Diagram* Kelola Stok |

#### State Diagram Pengajuan Stok

|  |
| --- |
|  |
| Gambar 3.12 *State Diagram* Pengajuan Stok |

#### State Diagram Penerimaan Stok

|  |
| --- |
|  |
| Gambar 3.13 *State Diagram* Penerimaan Stok |

### *Sequence Diagram*

*Sequence Diagram* merupakan bentuk kedua dari *behavioral model*. *Sequence Diagram* menunjukan bagaimana sebuah kejadian *(event)* menyebabkan transisi dari objek ke objek.

#### Sequence Diagram Alat Makan

|  |
| --- |
|  |
| Gambar 3.14 *Sequence Diagram* Alat Makan |

#### Sequence Diagram Kelola Order

|  |
| --- |
|  |
| Gambar 3.15 *Sequence Diagram* Kelola Order |

#### Sequence Diagram Kelola Stok

|  |
| --- |
|  |
| Gambar 3.16 *Sequence Diagram* Kelola Stok |

#### Sequence Diagram Pengajuan Stok

|  |
| --- |
|  |
| Gambar 3.17 *Sequence Diagram* Pengajuan Stok |

#### Sequence Diagram Penerimaan Stok

|  |
| --- |
|  |
| Gambar 3.18 *Sequence Diagram* Penerimaan Stok |

# BAB IV *DESIGN MODELLING*

*Design Modelling* membahas tentang gambaran dari sistem yang akan dibangun. *Design Modeling* meliputi *Architectural Design*, *User Interface Design*, *Component Level Design*, *Pattern Design*.



## *Architectural Design*

Desain Arsitektur mendefinisikan relasi antara struktur element utama yang ada didalam aplikasi, *style* arsitektur dan pola *design* yang dapat dimanfaatkan untuk mencapai definisi analisis yang telah ditetapkan sistem yang kemudian nanti dapat diimplementasikan. (Pressman, 2010).

### Struktur Menu

|  |
| --- |
|  |
| Gambar 4.1 Struktur Menu SIRELOKA |

## *User Interface Design*

Menggambarkan bagaimana *software* berkomunikasi dengan sistem dan manusia yang menggunakannya. Antarmuka menggambarkan aliran informasi dan prilakunya. (Pressman, 2010).

1. **Halaman Kelola Order**

|  |
| --- |
|  |
| Gambar 4.2 *User Interface Design* Halaman Kelola Order |

1. **Halaman Kelola Stok Bahan Makanan**

|  |
| --- |
|  |
| Gambar 4.3 *User Interface Design* Halaman Stok Bahan Makan |

1. **Halaman Peralatan Makan**

|  |
| --- |
|  |
| Gambar 4.4 *User Interface Design* Halaman Peralatan Makan |

1. **Halaman Pengajuan Stok**

|  |
| --- |
|  |
| Gambar 4.5 *User Interface Design* Halaman Pengajuan Stok |

1. **Halaman Penerimaan Stok**

|  |
| --- |
|  |
| Gambar 4.6 *User Interface Design* Halaman Penerimaan Stok |

## *Component Level Design*

*Component Level Design* menstranformasi elemen struktur dari arsitektur perangkat lunak kedalam deskripsi prosedural. Informasi didapatakan dari pemodelan kelas, pemodelan aliran data, dan pemodelan prilaku (Pressman, 2010).

### Algoritma

Algoritma adalah hasil produk *component level design* yang diambil dari method pada *Class diagram*.

|  |
| --- |
| ALGORITMA Edit Data Stok Bahan Makanan |
| DEKLARASI  kode\_makanan : string  nama\_makanan : string  jumlah\_makanan : integer  jenis\_makanan : string  tanggal\_expired : date |
| DESKRIPSI  Begin  Execute SQL "UPDATE STOK\_BAHAN\_MAKANAN(nama makanan, jumlah\_makanan,  jenis\_makanan, kode\_makanan) WHERE kode\_makanan='kondisi'");  If sql==true then  Output("Tampil Data");  Else  Output("Gagal");  End if  End |

|  |
| --- |
| ALGORITMA Terima Stok Bahan Makanan |
| DEKLARASI  kode\_makanan : integer  nama\_makanan : string  stok\_makanan : integer  jenis\_makanan : string  tanggal\_masuk : date  tanggal\_expired : date |
| DESKRIPSI  Begin  Input(kode\_makanan, nama\_makanan, stok\_makanan, jenis\_makanan, tanggal\_masuk, tanggal\_expired);  Execute SQL "INSERT INTO Penerimaan\_stok(kode\_am, nama\_am, jumlah\_am, kelayakan);  If sql==true then  Output("Input Data Berhasil");  Else  Output("Gagal");  End if  End |

|  |
| --- |
| ALGORITMA Hapus Data Stok Bahan Makanan |
| DEKLARASI  nama\_makanan : string  jumlah\_makanan : integer  jenis\_makanan : string  tanggal\_expired : date |
| Begin  Execute SQL “Select \* Form absensi WHERE *tanggal*=kondisi bulan”);  If sql == true then  Output(“Menampilkan Data”);  Else  Output(“Kosong”);  End if  End |

|  |
| --- |
| ALGORITMA Tambah Data Peralatan Makan |
| DEKLARASI  kode\_am : string  nama\_am : string  jumlah\_am : integer  kelayakan : string |
| DESKRIPSI  Begin  Input(kode\_am, nama\_am, jumlah\_am, kelayakan);  Execute SQL "INSERT INTO STOK\_BAHAN\_MAKANAN(kode\_am, nama\_am, jumlah\_am, kelayakan);  If sql==true then  Output("Input Data Berhasil");  Else  Output("Gagal");  End if  End |

|  |
| --- |
| ALGORITMA Edit Data Peralatan Makan |
| DEKLARASI  kode\_am : string  nama\_am : string  jumlah\_am : integer  kelayakan : string |
| DESKRIPSI  Begin  Execute SQL "UPDATE STOK\_BAHAN\_MAKANAN(nama\_am, jumlah\_am, kode\_am)  WHERE kode\_am='kondisi'");  If sql==true then  Output("Tampil Data");  Else  Output("Gagal");  End if  End |

|  |
| --- |
| ALGORITMA Hapus Data Makanan |
| DEKLARASI  kode\_am : string  nama\_am : string  jumlah\_am : integer  kelayakan : string |
| DESKRIPSI  Begin  Execute SQL "DELETE\*FROM STOK\_BAHAN\_MAKANAN WHERE kode\_am='kondisi'";  If sql==true then  Output("Data Berhasil di Hapus");  Else  Output("Gagal");  End if  End |

## *Deployment Level Design*

Desain level *deployment* mengindikasikan bagaimana aplikasi berfungsi dan subsistem dapat dialokasikan kedalam lingkungan terkomputerisasi dan akan mendukung sistem (Pressman, 2010).

|  |
| --- |
|  |
| Gambar 4.3. *Deployment Level Design* |

# BAB V PENUTUP

## Kesimpulan

Dari hasil penelitian disimpulkan bahwa Sistem Informasi merupakan salah satu hal yang terpenting di sebuah instansi perusahaan dalam hal ini yaitu restoran. Dengan adanya Sistem Informasi, instansi perusahaan dapat menjamin kualitas informasi yang disajikan. Seiring dengan perkembangan teknologi informasi, kebutuhan akan informasi yang cepat, tepat dan akurat sangat diperlukan. Karena itu, keberadaan sistem yang terkomputerisasi yang handal sudah menjadi kebutuhan mutlak bagi instansi dalam proses kerjanya.

Salah satu untuk mendukung sistem terkomputerisasi ini maka Sistem Informasi Pengelolaan Kitchen Restoran Sangriloka yang akan membantu segala pelayanan dalam hal proses pendataan di bagian kitchen restoran Sangriloka.

## Saran

Saran yang dapat diajukan terkait dengan pembuatan aplikasi Sistem Informasi Pengelolaan Kitchen Restoran Sangriloka adalah :

1. Akan lebih baik jika aplikasi ini lebih diperkuat lagi di sisi keamanan data, karena aplikasi ini masih bisa dibilang sederhana.
2. Pelatihan ke pegawai kitchen restoran Sangriloka tentang penggunaan aplikasi ini agar semua pegawai bisa menggunakan aplikasi.

# DAFTAR PUSTAKA

Amsyah, Z. (2005). *Manajemen Sistem Informasi.* Jakarta: Gramedia Pustaka Utama.

Bentley, Lonnie D dan Whitten, Jeffrey L. (2007). *System Analysis and Design for the Global Enterprise, 7th Edition, International Edition.* New York: McGrayHill.

Fowler, M. (2005). *UML Defined Edisi 3.* Yogyakarta: Andi.

Janner, S. (2010). *Rekayasa Perangkat Lunak.* Yogyakarta: Andi Offset.

Jogiyanto. (2005). *Analisis dan Desain Sistem Informasi.* Yogyakarta: Andi.

Kristanto, A. (2008). *Perancangan SIstem Informasi dan Aplikasinya.* Yogyakarta: Gava Media.

L. James, H. (2000). *Sistem Informasi .* Jakarta: Mata Satu.

LO’Brien, James A. dan Marakas, George M. (2011). *“Management InFormation Systems, 10th Edition”.* New York: McGraw-Hill/ Irwin.

Pressman, R. S. (2010). *“Rekayasa Perangkat Lunak”, Buku Satu. .* Yogyakarta: Andi.

Yasin, V. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek.* Jakarta: Mitra Wacana Media.

# LAMPIRAN

LAMPIRAN 1. LEMBAR *ASESSEMENT USER INTERFACE*

LAMPIRAN 2. *SOURCE CODE*

LAMPIRAN 3. ABSEN RESPONSI