

MEMAHAMI SISTEM SELF SERVICE MENGGUNAKAN UNIFIED MODELING LANGUAGE

Oleh:

Ho Lusi^{1*}, Syaiful Rahman², Abdul Munir³

^{1,2,3}Informatika, STMIK K HARISMA Makassar

e-mail: ¹holusi_18@kharisma.ac.id, ²syaifulrahman@kharisma.ac.id,

³abdulmunir@kharisma.ac.id

Abstrak: Penelitian ini bertujuan untuk mengetahui apakah dengan menggunakan Unified Modeling Language maka pengembang aplikasi dapat mudah memahami sistem self service. Metode yang digunakan dalam penelitian adalah metode Unified Modeling Language untuk menghasilkan pemodelan sistem dalam bentuk tiga buah diagram Unified Modeling Language (use case diagram, activity diagram, dan class diagram) dan metode tes untuk menghasilkan nilai rata-rata pemahaman responden tes sebesar 89,75. Berdasarkan hasil penelitian yang diperoleh, disimpulkan bahwa penggunaan Unified Modeling Language mempermudah pengembang aplikasi memahami sistem self service dari aplikasi Self Service in Food Menu and Order sehingga dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi Self Service in Food Menu and Order ke depannya.

Kata kunci: self service, Unified Modeling Language, UML, pemodelan, perancangan

Abstract: This study aims to determine whether by using the Unified Modeling Language application developers can easily understand the self service system. The method used in this study is the Unified Modeling Language method to produce system modeling in the form of three Unified Modeling Language diagrams (use case diagrams, activity diagrams, and class diagrams) and the test method to produce an average score of 89.75 test respondents' understanding. Based on the research results obtained, it is concluded that the use of the Unified Modeling Language makes it easier for application developers to understand the self service system of the Self Service in Food Menu and Order application so that it can be used as a benchmark if you want to develop Self Service in Food Menu and Order applications in the future.

Keywords: self service, Unified Modeling Language, UML, modeling

1. PENDAHULUAN

Selama ini saat berada di restoran, rumah makan, atau kafe, menu makanan dijumpai dalam bentuk fisik seperti penggunaan buku menu atau kertas menu serta pemesanan makanan yang diinginkan konsumen akan dilayani dan dicatat manual menggunakan kertas oleh pramusaji. Kadang kala kita dapat menjumpai pesanan yang diinginkan tidak sesuai, baik salah di-input atau tidak di-input. Untuk mengatasinya, penggunaan teknologi diperlukan untuk meningkatkan kualitas layanan.

Dengan menggunakan aplikasi *mobile Self Service in Food Menu and Order*, menu makanan akan disajikan secara digital berbasis aplikasi yang interaktif dan *user friendly*

* Corresponding author : Ho Lusi (holusi_18@kharisma.ac.id)

sehingga pemesanan makanan dapat langsung dilakukan sendiri oleh konsumen melalui aplikasi tersebut. Aplikasi tersebut diakses melalui perangkat *mobile* yang disediakan di setiap meja pelanggan oleh pihak rumah makan yang menggunakan layanan dari aplikasi ini. Penggunaan aplikasi ini membuat pihak rumah makan tidak perlu lagi menugaskan pramusaji untuk melayani dan mencatat pesanan pelanggan. Pelanggan hanya perlu menggunakan aplikasi ini, memesan darinya, dan memberikan catatan khusus terkait pesannya bila diperlukan.

Aplikasi ini bersifat *custom* yang artinya penggunaannya menyesuaikan dengan rumah makan yang menggunakan layanan ini sehingga penggunaannya hanya dalam ruang lingkup rumah makan tersebut. Namun, bentuk contoh dari aplikasi ini dapat ditemukan di play store, dengan nama *Self Service in Food Menu and Order*. Adapun link aplikasinya adalah sebagai berikut: <https://play.google.com/store/apps/details?id=id.kharisma.studio.selfservice>.

Aplikasi *Self Service in Food Menu and Order* menggunakan konsep teknologi *self service*. Sementara penggunaan konsep ini masih terbilang kurang dijumpai dalam lingkungan rumah makan, kafe, ataupun restoran sehingga pemahaman para pengembang aplikasi tentang alur kerja dari sistem yang menggunakan konsep teknologi *self service* masih terbilang kurang. Kurangnya pemahaman dapat menyebabkan sistem yang dirancang menjadi tidak tepat dan proses pengembangan aplikasi dapat terhambat sehingga diperlukan pemahaman mengenai konsep teknologi *self service* agar pengembangan dapat berjalan lancar. Untuk memahami alur kerja suatu sistem, dapat digunakan *Unified Modeling Language*.

Unified Modeling Language[1] sebagai standar penyajian model suatu sistem perangkat lunak, mendukung penggunaan empat belas jenis diagram UML. Akan tetapi, tidak semua jenis diagram tersebut digunakan. Berdasarkan prinsip pareto yang mengemukakan bahwa 80% hasil merupakan buah dari 20% usaha yang dilakukan, maka dalam penelitian ini digunakan 20% dari jumlah UML yang ada sehingga diperoleh tiga jenis diagram UML yang akan digunakan, yaitu *use case diagram*, *activity diagram*, dan *class diagram*. Selain itu, Sommerville[2] juga mengemukakan bahwa dengan meminimalkan jumlah model yang dihasilkan dapat mengurangi biaya desain dan waktu yang dibutuhkan untuk menyelesaikan proses desain. *Use case diagram*, *activity diagram*, dan *class diagram* termasuk ke dalam jenis diagram yang dapat mewakili esensi suatu sistem[2] dan jenis yang umum digunakan dalam pengembangan sistem[1], seperti penelitian terdahulu yang dilakukan oleh Fatmasari dan Sauda[3] serta Purnamasari dan Panjaitan[4] yang menggunakan ketiga jenis diagram tersebut.

Untuk memodelkan sistem, UML cocok untuk digunakan karena merupakan bahasa yang sangat ekspresif, namun tidak sulit untuk dipahami dan digunakan. Penggunaan UML menghasilkan model standar yang dapat ditafsirkan oleh semua pengembang tanpa memandang ia merupakan pengembang dari sistem tersebut atau bukan. Terkadang ada beberapa hal tentang sistem perangkat lunak yang tidak dapat dipahami melalui *source code* program, diperlukan gambaran grafis dan UML menjadi solusi karena merupakan sebuah

bahasa grafis. Dengan model eksplisit yang dihasilkan, komunikasi di antara pengembang dapat difasilitasi. Spesifikasi dari hasil analisis, desain, dan implementasi yang harus ada dalam pengembangan dan penyebaran suatu sistem dapat dispesifikasi dengan UML sehingga dapat membentuk model yang tepat, tidak ambigu, dan lengkap. Walau UML bukan bahasa pemrograman visual, tetapi modelnya dapat langsung dihubungkan ke berbagai bahasa pemrograman. Artinya, memungkinkan untuk memetakan dari model dalam UML ke bahasa pemrograman seperti Java, C++, Visual Basic, atau bahkan ke tabel dalam *database* relasional atau penyimpanan persisten dari *database* berorientasi objek. Adapun semua jenis artefak yang dihasilkan dari pengembangan sistem dapat pula didokumentasikan dengan UML[5].

Penelitian terkait yang dilakukan oleh Fatmasari dan Sauda[3] menghasilkan pemodelan yang diimplementasikan dalam bentuk *prototype* untuk memberikan gambaran jelas bagi pihak berkepentingan dalam pengembangan sistem informasi *Enterprise Resource Planning* (ERP) pada lingkungan PTPN VII. Sementara penelitian yang dilakukan peneliti hanya sebatas menghasilkan pemodelan yang tidak mengimplementasikan *prototype*, serta objek yang diteliti juga berbeda, yaitu sistem pemesanan makanan sendiri oleh konsumen pada aplikasi *Self Service in Food Menu and Order*. Selain itu, penelitian yang dilakukan peneliti menghasilkan nilai rata-rata 89,75 yang menunjukkan bahwa penggunaan UML mempermudah pengembang aplikasi memahami sistem *self service* dari aplikasi *Self Service in Food Menu and Order* sehingga dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi *Self Service in Food Menu and Order* ke depannya jika nilai rata-rata yang diperoleh responden sama dengan atau lebih dari 75.

Adapun Syarif dan Nugraha[6] melakukan penelitian yang bertujuan untuk memodelkan sistem pembayaran menggunakan diagram UML sehingga kebutuhan dan konten yang dikerjakan sistem tergambar jelas. Sementara penelitian yang dilakukan peneliti memiliki objek penelitian berbeda, yaitu sistem pemesanan makanan sendiri oleh konsumen pada aplikasi *Self Service in Food Menu and Order*, serta penelitian yang dilakukan peneliti menghasilkan nilai rata-rata 89,75 yang menunjukkan bahwa penggunaan UML mempermudah pengembang aplikasi memahami sistem *self service* dari aplikasi *Self Service in Food Menu and Order* sehingga dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi *Self Service in Food Menu and Order* ke depannya jika nilai rata-rata yang diperoleh responden sama dengan atau lebih dari 75.

Sementara itu, penelitian dari Purnamasari dan Panjaitan[4] memodelkan sistem informasi pemetaan pasar guna mendapatkan informasi yang benar bagi pihak terkait yang dibuktikan melalui *prototype* sistem tersebut. Sementara penelitian yang akan dilakukan peneliti hanya sebatas menghasilkan pemodelan yang tidak dibuktikan melalui *prototype*, serta objek yang diteliti juga berbeda, yaitu sistem pemesanan makanan sendiri oleh konsumen pada aplikasi *Self Service in Food Menu and Order*. Selain itu, penelitian yang dilakukan peneliti menghasilkan nilai rata-rata 89,75 yang menunjukkan bahwa penggunaan UML mempermudah pengembang aplikasi memahami sistem *self service* dari aplikasi *Self*

Service in Food Menu and Order sehingga dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi *Self Service in Food Menu and Order* ke depannya jika nilai rata-rata yang diperoleh responden sama dengan atau lebih dari 75.

Unified Modeling Language (UML) adalah bahasa pemodelan terstandarisasi yang terdiri dari kumpulan diagram terintegrasi yang digunakan untuk memvisualisasikan, menentukan, dan membangun artefak dari sistem perangkat lunak, ataupun pemodelan bisnis dan sistem non-perangkat lunak lainnya. *Unified Modeling Language* memungkinkan penggambaran sistem menggunakan kata dan gambar[7]. Sebagai proses pengembangan model abstrak dari suatu sistem, dengan masing-masing model menyajikan pandangan atau perspektif yang berbeda dari sistem tersebut, Sommerville[2] menyatakan bahwa sekarang dalam merepresentasikan sistem menggunakan semacam notasi grafis, hampir selalu didasarkan pada notasi dalam *Unified Modeling Language* (UML).

Fatmasari dan Sauda[3] menuliskan bahwa *Unified Modeling Language* merupakan alat perancangan sistem yang berorientasi objek. *Unified Modeling Language* didasari oleh konsep permodelan *Object Oriented* (OO), dikarenakan konsep ini merupakan sistem seperti kehidupan nyata yang didominasi oleh objek dan dinotasikan dalam simbol yang cukup spesifik sehingga *Object Oriented* (OO) memiliki proses standar dan bersifat independen. *Unified Modeling Language* dapat membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Notasi *Unified Modeling Language* diperoleh dari penyatuan tiga metode yang telah ada sebelumnya yaitu Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*), dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*).

Menurut Booch et al[5], pemodelan UML terutama digunakan dalam ruang lingkup sistem informasi perusahaan, layanan perbankan dan keuangan, telekomunikasi, transportasi, pertahanan/kedirgantaraan, eceran, elektronik medis, ilmiah, dan layanan berbasis web terdistribusi. Namun, penggunaannya tidak terbatas dalam pemodelan perangkat lunak. Bahkan, cukup ekspresif untuk memodelkan sistem non perangkat lunak, seperti alur kerja dalam sistem hukum, struktur dan perilaku sistem perawatan kesehatan pasien, dan desain perangkat keras. Untuk dapat melakukan pemodelan menggunakan UML, diperlukan model konseptual bahasa yang diperoleh dengan mempelajari tiga elemen utama: blok bangunan dasar UML, aturan yang menentukan bagaimana blok bangunan tersebut dapat disatukan, dan beberapa mekanisme umum yang berlaku di seluruh UML-nya.

UML versi 2.4.1 dalam tulisan Seidl et al[8] menetapkan adanya empat belas diagram yang terbagi ke dalam tiga kategori. Tujuh diagram menggambarkan struktur statis aplikasi, tiga diagram menggambarkan jenis perilaku secara rinci, dan empat diagram menggambarkan aspek interaksi yang berbeda. Berikut penggolongan kategori diagram-diagram tersebut.

a. Diagram Struktur (*Structure Diagram*)

Menggambarkan hubungan dan keterkaitan elemen di dalam sistem informasi, seperti struktur bentuk sistem informasi, termasuk ke bagian abstraksi dan implementasi sistem informasi[4]. *Class diagram, object diagram, component diagram, composite structure diagram, package diagram, deployment diagram, dan profile diagram* digolongkan dalam kategori *structure diagram*[8].

b. Diagram Perilaku (*Behavior Diagram*)

Purnamasari dan Panjaitan[4] menyatakan bahwa diagram kategori ini menggambarkan tingkah laku sebuah sistem informasi dan bagaimana sistem informasi tersebut melakukan tindakan terhadap kejadian atau perubahan. *Use case diagram, activity diagram, dan state machine diagram* digolongkan dalam kategori *behavior diagram*[8].

c. Diagram Interaksi (*Interaction Diagram*)

Merupakan turunan dari diagram perilaku[9] yang memberikan penekanan utama pada interaksi objek[10]. *Sequence diagram, communication diagram, timing diagram, dan interaction overview diagram* digolongkan dalam kategori *interaction diagram*[8].

Menurut Sommerville[2], terdapat lima jenis diagram UML yang sangat bermanfaat dalam memodelkan sistem, antara lain sebagai berikut.

- a. *Activity diagram*, yang menunjukkan aktifitas yang terlibat dalam sebuah proses atau dalam pemrosesan data.
- b. *Use case diagram*, yang menunjukkan interaksi antara sebuah sistem dan lingkungannya.
- c. *Sequence diagram*, yang menunjukkan interaksi antara para aktor dan sistem dan antara komponen sistem.
- d. *Class diagram*, yang menunjukkan kelas-kelas objek dalam sistem dan asosiasi antara kelas-kelas tersebut.
- e. *State diagram*, yang menunjukkan bagaimana sistem bereaksi dengan peristiwa internal dan eksternal.

Activity diagram merupakan diagram yang menggambarkan alur kerja atau aktifitas dari sebuah sistem yang ada pada perangkat lunak[11]. Fungsi utama diagram adalah untuk memodelkan logika lengkap dalam suatu sistem dan merupakan jenis diagram yang paling disukai pelajar[10].

Menurut Putra dan Andriani[11], *use case diagram* merupakan pemodelan untuk menjelaskan kelakuan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai.

Sommerville[2] menyatakan bahwa suatu *use case diagram* terdiri dari beberapa elemen, yakni aktor, *use case*, dan relasi. Setiap *use case* mewakili tugas diskrit yang melibatkan interaksi eksternal dengan sistem. *Use case diagram* hanya menunjukkan

gambaran interaksi sederhana sehingga dibutuhkan perincian untuk melengkapi interaksi tersebut. Perincian tersebut dapat berupa deskripsi tekstual, deskripsi terstruktur dalam tabel, atau *sequence diagram*.

Dalam tulisan Purnamasari dan Panjaitan[4], *class diagram* menggambarkan kelas pada sebuah sistem informasi dan hubungan antar kelas tersebut. Sementara itu, Putra dan Andriani[11] mendefinisikan *class diagram* sebagai gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kemudian, *class diagram* terdiri dari atribut dan operasi yang bertujuan agar para pembuat program dapat menghubungkan antara dokumentasi perancangan dan perangkat lunak sesuai.

Dengan demikian, penelitian ini bertujuan untuk mengetahui apakah dengan menggunakan *Unified Modeling Language* maka pengembang aplikasi (*application developer*) dapat mudah memahami sistem *self service*.

2. METODE PENELITIAN

Tahapan penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

a. Perumusan Masalah

Pada tahap ini, dilakukan pengamatan terhadap aplikasi rintisan tim *Founder* untuk mengidentifikasi masalah apa yang dimiliki aplikasi tersebut.

b. Tujuan Penelitian

Setelah masalah berhasil dirumuskan, ditetapkan tujuan dari penelitian yang akan dilakukan.

c. Pengumpulan Data

Data yang dikumpulkan adalah data kualitatif dan kuantitatif yang bersumber dari sumber data primer dan sekunder. Data primer diperoleh dari dokumen internal tim *Founder*, hasil observasi terhadap aplikasi rintisan tim *Founder*, dan responden tes. Sementara data sekunder diperoleh dari jurnal dan buku. Dalam tahap ini, pengumpulan data primer dilakukan melalui metode dokumen, observasi, dan tes. Sedangkan pengumpulan data sekunder dilakukan dengan cara studi literatur melalui jurnal dan buku tentang teori pemodelan *Unified Modeling Language*.

d. Pengolahan Data

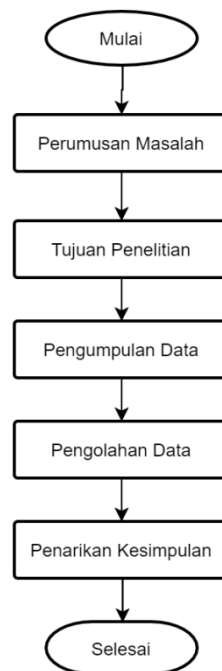
Metode yang digunakan dalam mengolah data adalah metode analisis. Data primer yang telah dikumpulkan dari hasil dokumen dan observasi akan diklasifikasi dan disusun berdasarkan teori pemodelan yang diperoleh. Kemudian, diterapkan untuk digambarkan ke dalam bentuk *use case diagram*, *activity diagram*, dan *class diagram* menggunakan *tools* Visual Paradigm. Sementara, data primer yang diperoleh dari tes akan dinilai jawabannya dan ditotalkan nilainya untuk memperoleh total nilai masing-masing responden tersebut.

e. Penarikan Kesimpulan

Penarikan kesimpulan dilakukan dengan menguji responden tes, responden tes akan diberikan gambar diagram UML yang dihasilkan dari penelitian ini. Setelah memahami gambar tersebut, mereka akan mengerjakan tes yang berisi pertanyaan menyerupai kuis

terkait diagram tersebut. Tes tersebut diisi oleh para responden menurut pemahamannya dan akan diberikan kembali ke peneliti untuk dinilai. Nilai yang diperoleh dari tes tersebut akan dikumpulkan dan dirata-ratakan untuk menunjukkan seberapa besar pemahaman mereka. Bila nilai rata-rata yang diperoleh di bawah 75, menunjukkan bahwa hipotesis dalam penelitian ini salah. Bila sama dengan atau di atas 75, menunjukkan bahwa hipotesis dalam penelitian ini benar. Nilai 75 didasarkan pada nilai KKM (Kriteria Ketuntasan Minimal) pada umumnya. Nilai rata-rata yang diperoleh akan diuji hipotesisnya menggunakan taraf signifikansi (α) 0,05 (paling banyak digunakan dalam penelitian pendidikan) dan tabel t.

Flow chart tahapan penelitian yang digunakan dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. *Flow Chart* Tahapan Penelitian

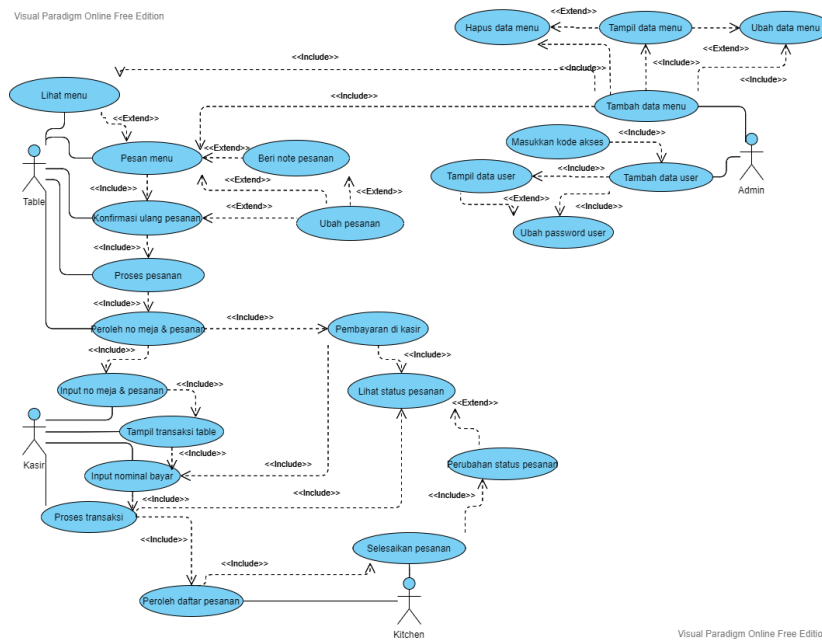
3. HASIL DAN PEMBAHASAN

3.1. Deskripsi Data

Berikut hasil yang diperoleh dalam penelitian ini.

- a. Diagram *Unified Modeling Language* dalam bentuk *use case diagram*

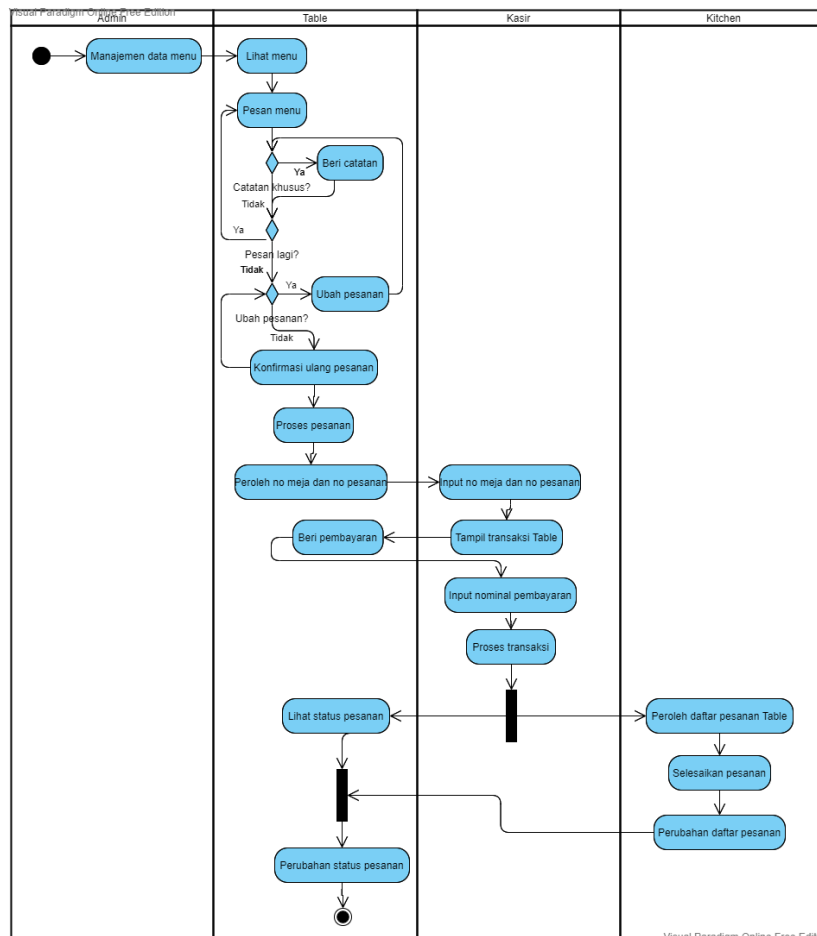
Dalam *use case diagram* yang dihasilkan, terdapat empat aktor yang terlibat sebagai pengguna sistem, yaitu Admin, Kasir, *Table*, dan *Kitchen*. Keempat aktor tersebut menunjukkan bagaimana interaksinya masing-masing dalam sistem yang dapat dilihat pada Gambar 2.



Gambar 2. Use Case Diagram

b. Diagram *Unified Modeling Language* dalam bentuk *activity diagram*

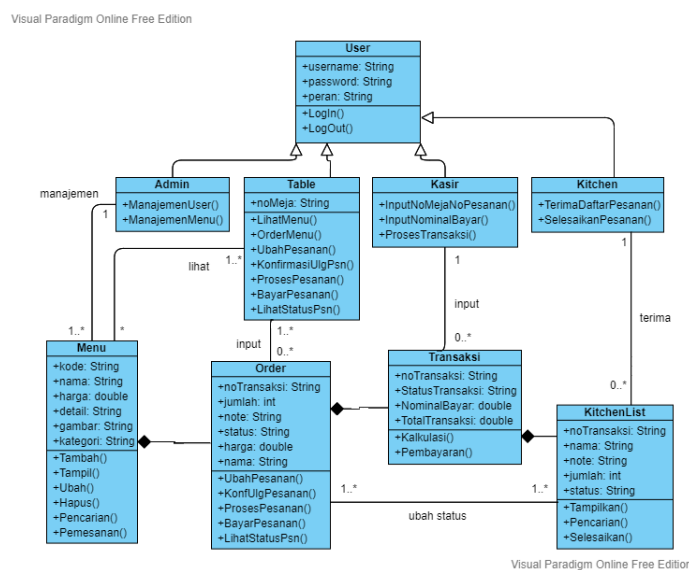
Sebagai bentuk penggambaran urutan aktifitas yang terjadi dalam sistem, dimulai dari aktifitas yang dilakukan Admin untuk *Table* agar proses pemesanan makanan pada *Table* dapat berlangsung hingga proses apa saja yang dilalui pesanan tersebut pada tahap Kasir dan Kitchen. Runtutan proses yang terjadi dapat dilihat pada Gambar 3.



Gambar 3. Activity Diagram

c. Diagram Unified Modeling Language dalam bentuk class diagram

Sistem digambarkan secara struktur dalam bentuk class dan hubungan diantaranya. Class yang terbentuk berisi nama, atribut, dan operasi dari class tersebut. Bentuk class diagram yang dihasilkan dapat dilihat pada Gambar 4.



Gambar 4. Class Diagram

d. Tabel nilai responden tes

Tabel nilai responden tes merupakan tabel hasil nilai yang diperoleh dari responden setelah mengerjakan tes soal pemahaman. Pada Tabel 1, i mewakili nomor responden (responden ke- i), jumlah responden (n) sebanyak 10 orang, x_i mewakili nilai yang diperoleh responden ke- i , dan \bar{x} merupakan nilai rata-rata yang diperoleh responden, yaitu 89,75. Nilai rata-rata diperoleh dengan menjumlahkan nilai yang diperoleh seluruh responden ($\sum_{i=1}^n(x_i)$), kemudian membaginya dengan jumlah responden (n).

Tabel 1: Tabel Nilai yang Diperoleh Responden Tes

RESPONDEN (i)	NILAI (x_i)
1	78,75
2	93,75
3	90
4	75
5	90
6	100
7	100
8	91,25
9	87,5
10	91,25
$n = 10$	$\bar{x} = 89,75$

3.2. Pembahasan

Dari tiga buah diagram UML yang dihasilkan, dilakukan pengujian tes pemahaman kepada sejumlah responden. Nilai yang diperoleh dari tes tersebut akan dikumpulkan dan dirata-ratakan untuk menunjukkan seberapa besar pemahaman mereka yang ditunjukkan melalui tabel 1. Dari tabel tersebut, diperoleh nilai rata-rata responden tes sebesar 89,75.

Pernyataan Hipotesis

H_0 = penggunaan UML mempermudah pengembang aplikasi memahami sistem *self service* dari aplikasi *Self Service in Food Menu and Order* sehingga dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi *Self Service in Food Menu and Order* ke depannya jika nilai rata-rata yang diperoleh responden sama dengan atau lebih dari 75.

H_a = penggunaan UML tidak mempermudah pengembang aplikasi memahami sistem *self service* dari aplikasi *Self Service in Food Menu and Order* sehingga tidak dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi *Self Service in Food Menu and Order* ke depannya jika nilai rata-rata yang diperoleh responden kurang dari 75.

Berikut formulasi hipotesisnya.

$H_0 : x = 75$

$H_a : x < 75$

dengan x merupakan nilai rata-rata.

Uji Hipotesis

Diketahui:

Jumlah responden = $n = 10$

Nilai rata-rata yang diperoleh responden = $\bar{x} = 89,75$

Nilai rata-rata pada hipotesis = $\mu_0 = 75$

Taraf signifikansi = $\alpha = 0,05$

$$\begin{aligned} \text{Simpangan baku} = S &= \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \\ &= \sqrt{\frac{574,375}{10-1}} \\ &= \sqrt{63,81944} \\ &= 7,988707 \end{aligned}$$

Kriteria pengujian:

Ho diterima jika $t_0 \geq -t_\alpha$

Ho ditolak jika $t_0 < -t_\alpha$

Nilai uji statistik:

$$\begin{aligned} t_0 &= \frac{\bar{x} - \mu_0}{\frac{S}{\sqrt{n}}} \\ &= \frac{89,75 - 75}{\frac{7,988707}{\sqrt{10}}} \\ &= \frac{14,75}{2,526251} \\ &= 5,838691 \\ -t_\alpha &= -1,833 \end{aligned}$$

4. KESIMPULAN

Dari hasil uji hipotesis, ditunjukkan bahwa $t_0 \geq -t_\alpha$ maka Ho diterima. Dengan demikian, hipotesis yang menyatakan bahwa penggunaan UML mempermudah pengembang aplikasi memahami sistem *self service* dari aplikasi *Self Service in Food Menu and Order* sehingga dapat digunakan sebagai patokan bila ingin mengembangkan aplikasi *Self Service in Food Menu and Order* ke depannya jika nilai rata-rata yang diperoleh responden sama dengan atau lebih dari 75 terbukti benar.

DAFTAR PUSTAKA

- [1] O. Filipova and O. Nikiforova, "Definition of the Criteria for Layout of the UML Use Case Diagrams," *Applied Computer Systems*, vol. 24, no. 1, pp. 75–81, May 2019, doi: 10.2478/acss-2019-0010.
- [2] I. Sommerville, *Software engineering*, 10th ed. London: Pearson.

- [3] F. Fatmasari and S. Sauda, "Pemodelan Unified Modeling Language Sistem Informasi Enterprise Resource Planning," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 2, p. 429, Apr. 2020, doi: 10.30865/mib.v4i2.2022.
- [4] S. D. Purnamasari, F. Panjaitan, J. Ahmad, Y. No3, and S. Ulu, "PEMODELAN SISTEM INFORMASI SEBARAN PASAR MENGGUNAKAN UNIFIED MODELING LANGUAGE," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 4, no. 2, pp. 103–110, 2019.
- [5] Grady. Booch, James. Rumbaugh, and Ivar. Jacobson, *The unified modeling language user guide*. Addison-Wesley, 1999.
- [6] M. Syarif and W. Nugraha, "PEMODELAN DIAGRAM UML SISTEM PEMBAYARAN TUNAI PADA TRANSAKSI E-COMMERCE," *Jurnal Teknik Informatika Kaputama (JTJK)*, vol. 4, no. 1, 2020.
- [7] A. A. Naing, M. T. Nyo, and S. M. Han, "An Analysis Design for Betel-Nut Business by Using Unified Modeling Language (UML)," *University Journal of Creativity and Innovative Research*, vol. 01, no. 1, pp. 114–118, 2020.
- [8] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, *Undergraduate Topics in Computer Science*. 2015. [Online]. Available: <http://www.springer.com/series/7592>
- [9] R. Diego De Oliveira, "Revista Brasileira de Design da Informação / Brazilian Journal of Information Design São Paulo | v," *Revista Brasileira de Design da Informação / Brazilian Journal of Information Design*, vol. 17, no. 1, pp. 116–130, 2020.
- [10] A. Alkhuwayr, "A Study of the State of the Art UML Case Tools Suitable for Supporting Teaching and Learning of Software Design and compatible with an ALL approach to the subject," *International Journal of Recent Research in Mathematics Computer Science and Information Technology*, vol. 6, no. 2, pp. 4–11, 2019, [Online]. Available: www.paperpublications.org
- [11] D. Wira, T. Putra, and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," vol. 7, no. 1, 2019.