

BAB 2

LANDASAN TEORI

2.1. Ruang Lingkup Penelitian

Objek penelitian yang sedang diteliti adalah perusahaan PT. Medika Khas Sadawira yang berada di Kota Bandung, yaitu bertempat di JL. Loncat Indah. No. 2, Kelurahan Sukamiskin, Kecamatan Arcamanik.

2.2. Sejarah Perusahaan

PT. Medika Khas Sadawira adalah perusahaan yang bergerak di bidang medis, khususnya di bidang farmasi dan juga alat - alat laboratorium kedokteran. yang didirikan pada 11 November 2021.

2.3. Visi Dan Misi PT. Medika Khas Sadawira

PT. Medika Khas Sadawira mempunyai visi dan misi sebagai berikut.

2.3.1. Visi PT. Medika Khas Sadawira

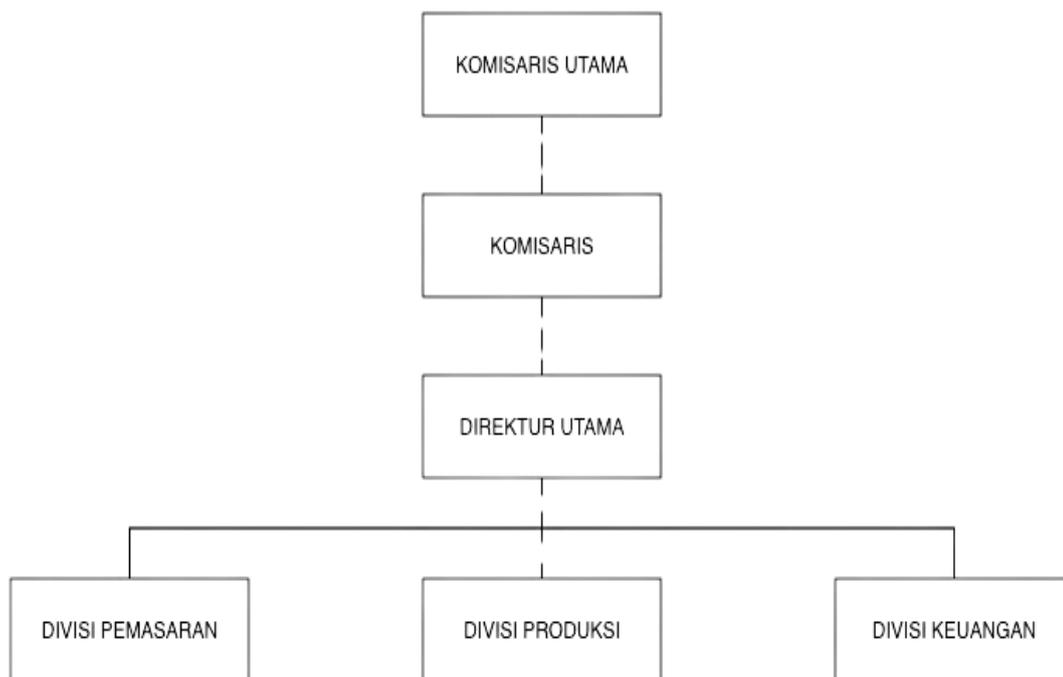
Menjadi penyelenggara layanan kesehatan terpercaya di Indonesia dan turut berkontribusi bagi peningkatan derajat kesehatan masyarakat.

2.3.2. Misi PT. Medika Khas Sadawira

1. Menjadi perusahaan terpercaya dalam memberikan layanan dan solusi.
2. Selalu berusaha meningkatkan mutu layanan.
3. Memenuhi asas kepatuhan pada regulasi yang berlaku.
4. Meningkatkan SDM yang kompeten dan profesional.

2.4. Struktur Organisasi PT. Medika Khas Sadawira

Berikut adalah struktur organisasi yang ada di PT. Medika Khas Sadawira yang di jelaskan pada gambar 2.1.



Gambar 2.1 Struktur Organisasi PT. Medika Khas Sadawira

Berikut adalah rincian fungsi dari masing-masing jabatan yang di jelaskan pada tabel 2.1 :

Tabel 2.1 Fungsi-fungsi dari jabatan

No	Jabatan	Keterangan
1	Komisaris Utama	Komisaris Utama memiliki peran utama dalam mengawasi dan memberikan arahan strategis kepada direksi. Mereka bertugas memastikan bahwa kebijakan dan strategi yang diterapkan oleh direksi sejalan dengan visi jangka panjang perusahaan dan kepentingan pemegang saham.

2	Komisaris	Anggota dewan komisaris yang bertugas mengawasi jalannya perusahaan secara keseluruhan. Mereka memberikan masukan, kritik, dan saran kepada direksi terkait kebijakan dan manajemen perusahaan. Komisaris juga memastikan bahwa perusahaan berjalan sesuai dengan peraturan perundang-undangan dan standar etika yang berlaku.
3	Direktur Utama	Sebagai pemimpin eksekutif tertinggi di perusahaan, Direktur Utama bertanggung jawab atas pengambilan keputusan strategis dan operasional. Mereka mengarahkan semua kegiatan perusahaan, menetapkan visi dan misi, serta memastikan bahwa tujuan perusahaan tercapai. Direktur Utama juga berperan dalam mewakili perusahaan di hadapan publik, pemegang saham, dan pihak-pihak eksternal lainnya.
4	Divisi Pemasaran	Divisi ini bertugas merancang dan mengimplementasikan strategi pemasaran yang bertujuan untuk meningkatkan penjualan, membangun brand awareness, dan memperkuat posisi perusahaan di pasar. Tanggung jawab mereka meliputi analisis pasar, pengembangan produk, promosi, manajemen hubungan pelanggan, dan pengelolaan saluran distribusi.

5	Divisi Produksi	Bertanggung jawab atas seluruh proses produksi barang atau jasa yang dihasilkan oleh perusahaan. Divisi ini memastikan bahwa proses produksi berjalan dengan efisien, tepat waktu, dan sesuai dengan standar kualitas yang telah ditetapkan. Mereka juga mengelola sumber daya, termasuk bahan baku dan tenaga kerja, serta melakukan perbaikan terus-menerus untuk meningkatkan produktivitas dan efisiensi.
6	Divisi Keuangan	Divisi ini mengelola semua aspek keuangan perusahaan, termasuk penyusunan anggaran, pengelolaan arus kas, akuntansi, dan pelaporan keuangan. Mereka bertanggung jawab untuk memastikan stabilitas finansial perusahaan, melakukan analisis keuangan untuk mendukung pengambilan keputusan strategis, serta memastikan kepatuhan terhadap regulasi keuangan dan perpajakan.

2.5. Media Sosial

Media sosial dapat dipahami sebagai suatu platform digital yang menyediakan fasilitas untuk melakukan aktivitas sosial bagi setiap penggunanya. Beberapa aktivitas yang dapat dilakukan di media sosial, misalnya yaitu melakukan komunikasi atau interaksi hingga memberikan informasi atau konten berupa tulisan, foto dan video [5].

Pesatnya perkembangan teknologi internet dan *mobile* menjadikan media sosial tumbuh dengan pesat pula. Kini mengakses media sosial bisa dilakukan kapan saja dan dimana saja, hal ini menyebabkan terjadinya fenomena besar terhadap arus informasi tidak hanya di negara-negara maju saja, tetapi juga di Indonesia. Tidak hanya itu, media sosial juga menjadi salah satu alat promosi bisnis yang sangat efektif. Media sosial menjadi bagian yang sangat diperlukan

oleh banyak perusahaan untuk melakukan pemasaran dan merupakan salah satu cara terbaik untuk menjangkau pelanggan dan klien.

2.6. Omnichannel

Omnichannel adalah strategi pemasaran yang memungkinkan perusahaan untuk berkomunikasi dengan pelanggan melalui berbagai platform media sosial, seperti *Facebook*, *Instagram*, dan *WhatsApp*, dalam satu platform terintegrasi. Tujuan utama dari strategi ini adalah untuk memberikan pengalaman pelanggan yang mulus dan konsisten di seluruh saluran [6].

Dalam penerapan strategi omnichannel, perusahaan harus dapat mengintegrasikan semua saluran komunikasi dan data pelanggan menjadi satu platform terpusat. Dengan cara ini, pelanggan dapat berinteraksi dengan merek tanpa merasa seperti mereka harus berbicara dengan departemen yang berbeda atau melalui saluran yang berbeda-beda [1].

Contoh untuk penerapan omnichannel di bidang ritel adalah sebagai berikut. Pelanggan dapat berbelanja secara online melalui situs web atau aplikasi mobile, dan kemudian mengambil barang di toko fisik, atau sebaliknya. Informasi mengenai stok, harga, dan promosi konsisten di semua saluran baik itu di situs web atau pun di aplikasi mobile.

2.7. Aplikasi

Secara harfiah, aplikasi adalah suatu penerapan perangkat lunak atau software yang dikembangkan untuk tujuan melakukan tugas-tugas tertentu.

Aplikasi juga dapat diartikan sebagai sebuah perangkat lunak atau program yang diciptakan dan dikembangkan untuk melakukan tugas-tugas tertentu pada perangkat komputer, laptop, atau ponsel [7].

2.8. Pemograman Berbasis Objek

Object Oriented Programming (OOP) atau dalam bahasa Indonesia disebut dengan Pemograman Berbasis Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan

objek yang berisi data dan operasi yang diberlakukan terhadapnya. *Object Oriented Programming* atau yang disingkat OOP ialah paradigma atau teknik pemrograman yang berorientasi kepada objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini sangat beragam (aplikasi bisnis, real-time, utility, dan sebagainya) dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut [8].

2.9. Basis Data

Basis data adalah penyimpanan kumpulan informasi secara sistematis dalam sebuah komputer sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (Database Management System, DBMS) [9].

2.10. Hypertext Preprocessor (PHP)

PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web [10].

Pada tahun 1995, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1995, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessor.

PHP adalah salah satu bahasa pemrograman open source yang sangat

cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skrip HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa scripting server – side, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan [11].

2.11. JavaScript

JavaScript adalah bahasa skrip yang digunakan untuk membuat konten halaman web dinamis, berfungsi untuk membuat elemen yang mampu meningkatkan interaksi pengunjung seperti menu drop-down, animasi, dan warna background dinamis [12].

JavaScript tidak memerlukan kompiler untuk menjalankannya, cukup dengan interpreter. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan. Browser Web Netscape Navigator dan Internet Explorer adalah salah satu interpreter, karena kedua browser ini telah dilengkapi dengan interpreter JavaScript. Tetapi tidak semua browser Website dapat menjadi interpreter JavaScript karena belum tentu browser tersebut dilengkapi dengan interpreter JavaScript.

2.12. NodeJs

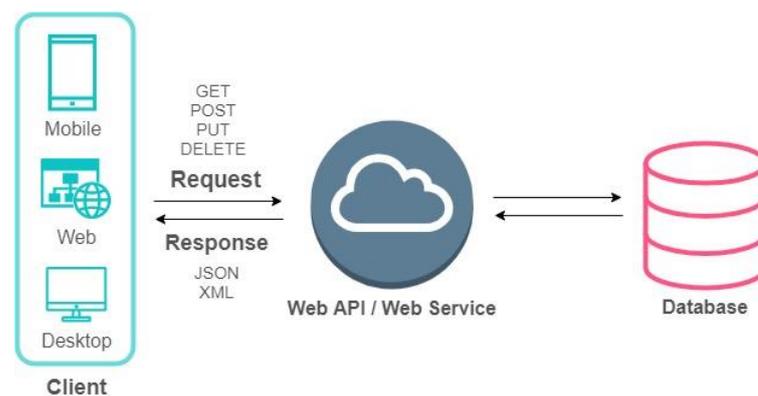
Node JS adalah software yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam Bahasa pemrograman javascript. Node JS melengkapi fungsi JavaScript yang biasanya berjalan di sisi client, namun Node JS juga dapat berjalan di sisi server [13]. Pada dasarnya, Node JS adalah sebuah runtime environment dan script library. Runtime environment yang mengeksekusi, menjalankan, dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu Bahasa pemrograman. Script library yang mengumpulkan, kompilasi, atau bank data berisi data kode pemrograman. Node JS menggunakan Bahasa JavaScript dan C, itu memiliki arsitektur dan fungsi Google V8, yang berfungsi sebagai compiler C, dan library Libuv menangani operasi I/O

asynchronous dan loop peristiwa utama. Node JS merupakan sistem yang bekerja secara asynchronous, yakni mampu menjalankan beberapa proses secara bersamaan. Proses runtime yang berjalan secara asynchronous ini sengaja dibuat agar mampu mendukung pengembangan aplikasi secara berkala.

2.13. *Application Programming Interface (API)*

Application Programming Interface (API) adalah sekumpulan aturan dan mekanisme yang memungkinkan berbagai aplikasi perangkat lunak untuk berkomunikasi satu sama lain. API memungkinkan pengembang untuk menggunakan fungsionalitas atau layanan yang disediakan oleh aplikasi atau platform lain tanpa perlu mengetahui detail implementasinya secara mendalam [14].

API digunakan untuk mempercepat pengembangan dengan memberikan fitur terpisah sehingga pembangun tidak perlu membuat fitur serupa. API biasanya berfungsi sebagai sumber data untuk kebutuhan sistem atau aplikasi tertentu. Ini memungkinkan pertukaran data dari satu aplikasi ke aplikasi lainnya [15]. Gambar 2.2 berikut menunjukkan cara kerja API.



Gambar 2.2 Cara Kerja Api

2.14. *Laravel*

Laravel adalah sebuah *framework* pengembangan web berbasis PHP yang telah muncul sebagai salah satu pemimpin dalam industri pengembangan

perangkat lunak. Dirancang dan dikembangkan oleh Taylor Otwell, *Laravel* menawarkan pendekatan yang bersih, ekspresif, dan efisien untuk membangun aplikasi web modern. Berikut adalah deskripsi lebih rinci mengenai fitur-fitur utama dan keunggulan yang membuat *Laravel* menjadi pilihan populer di kalangan pengembang [16][17].

Laravel menonjol dengan ekosistem yang kaya dan komunitas yang besar. Dengan adanya banyak paket *eksternal*, *library*, dan sumber daya online, pengembang dapat dengan mudah mengatasi tantangan pengembangan dan memperoleh dukungan dari komunitas aktif [16].

Secara keseluruhan, *Laravel* bukan hanya sekadar *framework*, tetapi juga sebuah ekosistem lengkap yang mempercepat proses pengembangan, meningkatkan kualitas kode, dan memberikan solusi untuk berbagai tantangan pengembangan web modern.

2.15. MySQL

MySQL adalah sebuah sistem manajemen basis data relasional (RDBMS) yang sangat populer. RDBMS adalah jenis database yang mengorganisir data menjadi tabel yang terdiri dari baris dan kolom, dan tabel-tabel ini dapat saling terkait. *MySQL* dikembangkan oleh perusahaan Swedia, *MySQL AB*, yang kemudian diakuisisi oleh *Sun Microsystems*. *Sun Microsystems* sendiri kemudian diakuisisi oleh *Oracle Corporation*, sehingga saat ini *MySQL* merupakan produk yang dimiliki dan didukung oleh *Oracle* [18].

MySQL telah diterapkan pada banyak aplikasi yang sudah populer di dunia. Salah satunya adalah aplikasi seperti *Facebook* yang memiliki skala pengguna yang besar.

2.16. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML menyediakan serangkaian diagram yang dapat digunakan untuk menggambarkan berbagai aspek dari sistem, termasuk struktur, perilaku,

dan interaksinya. Tujuan utama UML adalah untuk menyediakan standar untuk pemodelan sistem perangkat lunak.

Dengan tujuan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan sistem, UML mencakup konsep semantik, notasi, dan pedoman yang bersifat universal. UML mendefinisikan struktur statis dan perilaku dinamis suatu sistem, memungkinkan pemodelan dari berbagai sudut pandang yang terpisah namun terkait. Selain itu, UML juga menyediakan konstruksi organisasi untuk mempartisi sistem besar, mengelola ketergantungan, dan mewakili keputusan implementasi.

UML bukanlah bahasa pemrograman. Alat ini dapat menyediakan generator kode dari UML ke dalam berbagai bahasa pemrograman, serta membangun model rekayasa balik dari program yang sudah ada. UML bukanlah bahasa formal yang dimaksudkan untuk pembuktian teorema. Ada sejumlah bahasa seperti itu, namun tidak mudah dipahami atau digunakan untuk sebagian besar tujuan. UML adalah bahasa pemodelan tujuan umum. UML adalah bahasa pemodelan diskrit. Hal ini tidak dimaksudkan untuk memodelkan sistem kontinu seperti yang ditemukan dalam bidang teknik dan fisika. UML dimaksudkan untuk menjadi bahasa pemodelan tujuan umum universal untuk sistem diskrit seperti yang dibuat dari perangkat lunak, firmware, atau logika digital [19]. Dalam UML terdapat banyak diagram, empat diantaranya adalah sebagai berikut:

2.11.1. Use Case Diagram

Use Case Diagram merupakan jenis diagram *Unified Modeling Language* (UML) yang digunakan untuk menunjukkan hubungan antara sistem dan aktor. Jenis interaksi apapun yang dilakukan oleh pengguna sistem dengan sistem dapat digambarkan oleh *use case diagram*. *Use case diagram* digunakan untuk menggambarkan fungsionalitas sistem dari perspektif pengguna [19]. Diagram ini dapat membantu untuk memahami kebutuhan pengguna dan untuk mendokumentasikan persyaratan sistem. Simbol-simbol pada *use case* dapat dilihat pada Tabel 2.2 berikut.

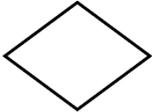
Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Keterangan
1		Aktor : Mewakili peran orang, proses, sistem yang lain atau alat berkomunikasi dengan <i>use case</i> .
2		<i>Use Case</i> : abstraksi dan interaksi antara sistem dan aktor.
3		<i>Association</i> : abstraksi dari penhubung antara aktor dengan <i>use case</i> .
4		<i>Generalisasi</i> : Menggambarkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		<i>Include</i> : menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6		<i>Extend</i> : Menunjukkan bahwa suatu <i>use case</i> merupakan tambahin fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2.11.2. Activity Diagram

Activity diagram merupakan jenis diagram yang digunakan untuk memodelkan proses bisnis, alur kerja, atau aktivitas dalam suatu sistem. *Activity* diagram membantu memahami, merancang, dan mendokumentasikan langkah-langkah yang terlibat dalam suatu aktivitas atau proses. Diagram ini membantu menggambarkan urutan aktivitas dan alur kerja dari suatu proses atau tugas. Karena memodelkan urutan dari satu aktivitas ke aktivitas lain atau dari aktivitas ke status, diagram ini mirip dengan *flowchart*. Membuat *activity* diagram pada awal pemodelan proses sangat membantu untuk memahami keseluruhan proses [19]. Simbol-simbol pada *activity* diagram dapat dilihat pada Tabel 2.3 berikut.

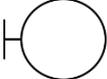
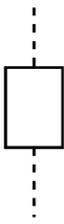
Tabel 2.3 Simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		Status awal	Menunjukkan awal dari sebuah proses
2		Aktivitas	Menggambarkan aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		Percabangan / Decision	Percabangan Dimana ada pilihan lebih dari satu aktivitas.
4		Penggabungan / Join	Penggabungan Dimana aktivitas yang lebih dari satu digabungkan menjadi satu aktivitas.
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.11.3. Sequence Diagram

Sequence diagram digunakan untuk menjelaskan dan menunjukkan interaksi antar objek dalam sebuah sistem. *Sequence* diagram juga menunjukkan pesan atau perintah yang dikirim, serta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan. Tujuan utama pembuatan *sequence* diagram adalah untuk mengetahui urutan kejadian yang dapat menghasilkan hasil yang diinginkan [19]. Contoh simbol-simbol *sequence* diagram dapat dilihat pada Tabel 2.4 berikut.

Tabel 2.4 Simbol Sequence Diagram

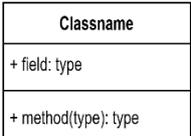
No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang, user, sistem yang lain atau alat berkomunikasi dengan sistem.
2		<i>Entity Class</i>	Menandakan hubungan yang akan dilakukan.
3		<i>Boundary Class</i>	Menandakan hubungan suatu <i>form, report, hardware interface</i> .
4		<i>Control Class</i>	Menandakan penghubung dan <i>boundary</i> dengan tabel.
5		<i>A Focus of Control and Lifetime</i>	Menandakan tempat mulai dan berakhirnya pesan.
6		<i>Message</i>	Menandakan pengiriman pesan.

2.11.4. Class Diagram

Class Diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan struktur, deskripsi class, atribut, metode, dan hubungan setiap objek. *Class diagram* memberikan pandangan terperinci tentang kelas-kelas (*class*) yang ada dalam sistem, hubungan antara kelas-kelas tersebut, atribut-atribut yang dimiliki oleh kelas, dan metode-metode yang dapat diakses oleh kelas. Namun, *class diagram* tidak menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, tetapi hanya menunjukkan hubungan yang terjadi [19]. Contoh simbol-simbol *class diagram*

dapat dilihat pada Tabel 2.5 berikut.

Tabel 2.5 Simbol Class Diagram

No	Simbol	Nama	Keterangan
1		<i>Class, field / attribute, method,</i>	<p><i>Class</i> atau kelas mempresentasikan suatu entitas atau objek dalam sistem.</p> <p><i>Field</i> atau <i>attribute</i> merupakan data atau <i>property</i> yang dimiliki oleh kelas.</p> <p><i>Method</i> merupakan fungsi atau operasi yang dimiliki oleh kelas.</p>
2		<i>Association</i>	Menggambarkan hubungan antar kelas, biasanya disertai dengan <i>multiplicities</i> .
3		<i>Generalization</i>	Menggambarkan hubungan antar sub kelas dengan super kelas.