

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Pada penelitian ini peneliti membuat judul penelitian "Sistem Informasi Akademik Berbasis Web pada SMA PGRI 1 BANDUNG". Penelitian terdahulu yang peneliti cantumkan dibawah ini merupakan acuan untuk dapat memperkaya teori peneliti dalam mengkaji penelitian yang dilakukan. Adapun penelitian yang peneliti kaji sebelumnya sebagai berikut.

Dalam jurnal yang dibuat oleh Destriady, Doddy dan R. Fenny Syafarani (2017) [2] dalam penelitian tersebut bertujuan menyelesaikan permasalahan yang terjadi pada SMP Angkasa Lanud Husein Sastranegara Bandung yang masih menggunakan proses manual. Sistem Informasi yang dibangun pada penelitian tersebut berbasis desktop yang memiliki fitur untuk melayani pendaftaran siswa baru, pembagian kelas, penjadwalan kelas, pembayaran spp serta pembuatan laporan keuangan. Bahasa pemograman yang digunakan adalah Java serta menggunakan database mysql. Pada penelitian ini menggunakan metode penelitian deskriptif dan menggunakan metode pengembangan sistem *prototype*.

Dalam Penelitian jurnal yang dilakukan oleh Wibawa, Julian Chandra (2017)[3] dalam penelitian ini bertujuan merancang dan menghasilkan sebuah sistem informasi berbasis web yang dapat menangani pengolahan data akademik meliputi; pendaftaran peserta didik baru, pengolah data nilai dan absensi yang nantinya dapat terintegrasi dengan nilai raport. Dengan adanya sistem ini diharapkan dapat menimalisir resiko terjadinya kesalahan.

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Tujuan Penelitian	Persamaan dan Perbedaan
----	---------------	------------------	-------------------	-------------------------

1	Destriady, Doddy dan R. Fenny Syafarani	Sistem Informasi Akademik SMP Angkasa Lanud Husein Sastranegara Bandung	Menyelesaikan permasalahan yang terjadi pada SMP Angkasa Lanud Husein Sastranegara Bandung yang masih menggunakan proses manual	<p>Persamaan :</p> <p>Menggunakan metode penelitian deskriptif dan membahas penjadwalan dan pembagian kelas</p> <p>Perbedaan :</p> <p>Pada penelitian ini membahas pengolahan data akademik yang ada pada tingkat smp, membahas membuat laporan untuk kepala sekolah, dan juga terdapat pendaftaran</p>
2	Wibawa, Julian Chandra	Rancang Bangun Sistem Informasi Akademik (Studi Kasus: Smpit Nurul Islam Tengaran)	Bertujuan merancang dan menghasilkan sebuah sistem informasi berbasis web yang mampu menangani pengolahan	<p>Persamaan :</p> <p>Sama-sama dalam membahas pengintegrasian nilai dan absensi untuk menghasilkan raport</p>

			data akademik meliputi; pendaftaran peserta didik baru, pengolahan data nilai dan absensi yang nantinya dapat terintegrasi dengan nilai raport	<p>Perbedaan :</p> <p>Pada penelitian metode pendekatan yang digunakan adalah terstruktur serta terdapat layanan untuk melayani pendaftaran peserta didik baru</p>
--	--	--	--	---

2.2. Teori Dasar

2.2.1. Pengertian Sistem

Menurut Yulia Djahir dan Pratita (2015:7), mengemukakan bahwa “Sistem adalah kumpulan atau grup dari subsistem atau bagian atau komponen apapun baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai tujuan tertentu” [4].

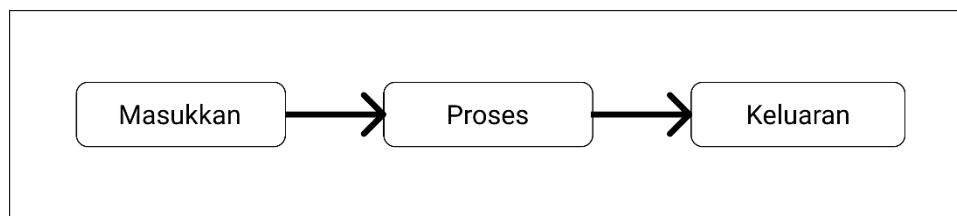
Sedangkan menurut Sri Mulyani (2016:2), menyatakan bahwa “Sistem bisa diartikan sebagai sekumpulan sub sistem, komponen yang saling bekerja sama dengan tujuan yang sama untuk menghasilkan Output yang sudah ditentukan sebelumnya” [5].

Dan terakhir menurut Jeperson Hutahaean (2014:2), berpendapat bahwa “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu” [6].

Suatu sistem termasuk sistem yang baik ketika mempunyai karakteristik berikut ini [6]:

1. Memiliki komponen-komponen yang saling bekerja sama membentuk satu kesatuan.
2. Memiliki batasan sistem (*boundary*) untuk memperlihatkan ruang lingkup (*scope*) dari sistem.
3. Mampu menjaga dan mengendalikan lingkungan luar sistem (*environment*) yang bersifat menguntungkan dan merugikan agar tidak mengganggu keberlangsungan suatu sistem.
4. Memiliki penghubung sistem (*interface*) untuk mengirim masukan atau keluaran dari subsistem ke subsistem lain.
5. Memiliki masukan (*input*) baik berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*) seperti data.
6. Memiliki keluaran (*output*) baik berupa keluaran yang berguna seperti informasi dan sisa pembuangan seperti panas.
7. Berfungsi sebagai pengolah untuk merubah masukan menjadi keluaran.
8. Memiliki sasaran atau tujuan agar dapat menentukan masukan yang diperlukan dan keluaran yang dihasilkan.

Berdasarkan karakteristik-karakteristik di atas dapat disimpulkan bahwa gambaran suatu sistem sebagai berikut:



Gambar 2. 1 Gambaran Suatu Sistem

(Sumber : Konsep Sistem Informasi [6])

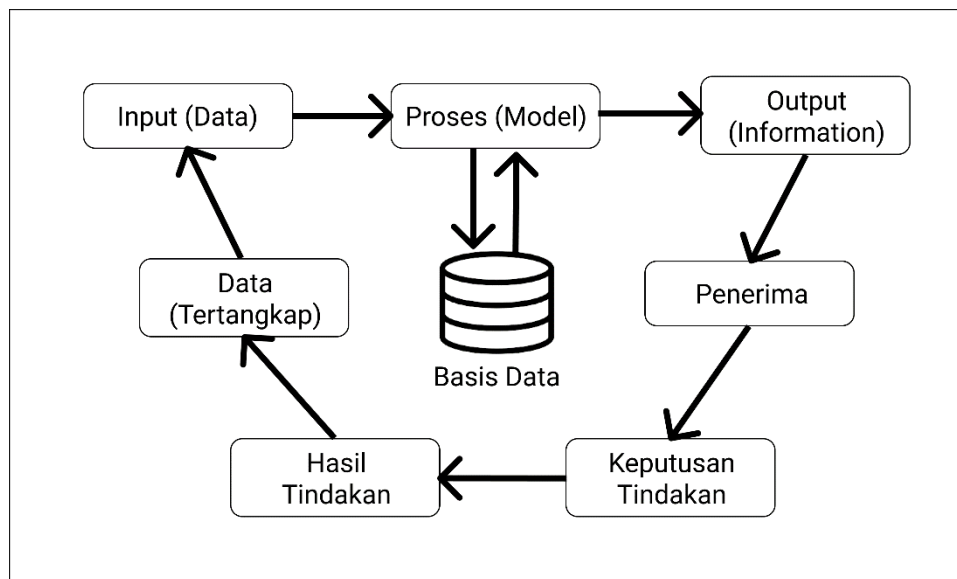
2.2.2. Pengertian Informasi

Informasi adalah hasil pengolahan data yang memberikan sesuatu yang lebih bermanfaat dan lebih bermakna bagi penerimanya [6]. Seorang ahli bernama Gordon B. Davis menambahkan bahwa informasi merupakan sesuatu yang memiliki nilai yang nyata yaitu dapat digunakan untuk memutuskan sesuatu pada saat ini maupun pada masa yang akan datang [6].

Pada umumnya informasi digunakan untuk menambah wawasan atau memberikan standard dalam pengambilan keputusan, dan informasi juga dapat membantu mempercepat pengambilan keputusan karena dapat memberikan gambaran suatu persoalan. Informasi akan sangat berguna ketika dalam kondisi yang tepat, berikut ini faktor-faktor yang mempengaruhi kebergunaan informasi [6]:

1. Informasi harus sesuai dengan kebutuhan penerimanya sehingga dapat membantu permasalahan penerimanya.
2. Setelah mengalami suatu pengolahan lalu menghasilkan suatu informasi maka inti dan pentingnya informasi tersebut tidak boleh hilang.
3. Informasi yang diberikan tidak boleh kadaluarsa artinya waktu terkini.
4. Informasi harus berada dalam tempat yang tepat.
5. Bentuk informasi harus sesuai dengan permasalahan yang ada.
6. Kata-kata dan arti dari informasi harus jelas sehingga tidak ada kemungkinan salah tafsir.

Untuk mendapatkan suatu informasi tertentu maka dibutuhkan suatu data tertentu. Data tersebut akan dijadikan sebagai masukan, lalu diproses menggunakan suatu model tertentu. Setelah itu menghasilkan suatu informasi yang dijadikan sebagai keluaran. Secara garis besar, itu yang dinamakan dengan siklus informasi (*information cycle*) atau bisa disebut juga siklus pengolahan data (*data processing cycles*). Siklus ini dapat dilihat pada gambar di bawah ini.



Gambar 2. 2 Siklus Informasi (*Information Cycle*)
(Sumber : Konsep Sistem Informasi [6])

Suatu informasi akan memiliki nilai dari informasi tersebut. Nilai informasi (Value of Information) tersebut bergantung pada dua hal yaitu tingkat kebergunaan dan jumlah biaya yang dikeluarkan. Nilai informasi akan tinggi ketika manfaatnya lebih besar dibandingkan dengan biaya yang dikeluarkan.

Berikut ini beberapa biaya yang mungkin dibutuhkan untuk mendapatkan suatu informasi [6]:

1. Biaya perangkat keras merupakan suatu biaya tetap atau tertanam.
2. Biaya analisis merupakan suatu biaya tetap atau tertanam.
3. Biaya untuk tempat dan faktor kontrol lingkungan merupakan suatu biaya setengah berubah atau semivariabel.
4. Biaya perubahan merupakan suatu biaya tetap atau tertanam contohnya perubahan satu metode ke metode lain.
5. Biaya operasi merupakan suatu biaya yang berubah-ubah atau biaya variabel contohnya biaya pegawai, pemeliharaan fasilitas, dan sistem.

2.2.3. Pengertian Sistem Informasi

Menurut seorang ahli bernama Alter pada tahun 1992, sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi [7]. Sedangkan menurut Turban, McLean, dan Wetherne tahun 2000, sistem informasi adalah sebuah sistem yang berfungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan spesifik [8].

Selain dua pendapat ahli di atas, masih ada definisi sistem informasi menurut para ahli lainnya. Namun, pada buku Pengenalan Sistem Informasi Edisi Revisi karya Abdul Karim (2014:8), dari berbagai definisi tersebut dapat disimpulkan bahwa sistem informasi adalah gabungan dari sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), terjadi proses pengolahan data menjadi informasi, dan informasi tersebut digunakan untuk mencapai suatu tujuan [9].

Sebenarnya, sistem informasi tidak harus melibatkan komputer. Namun pada kenyataannya, sebagian besar sistem informasi menggunakan komputer sebagai alat untuk mengimplementasi sistemnya. Sistem informasi yang menggunakan komputer disebut sistem informasi berbasis komputer (Computer Based Information Systems atau CBIS).

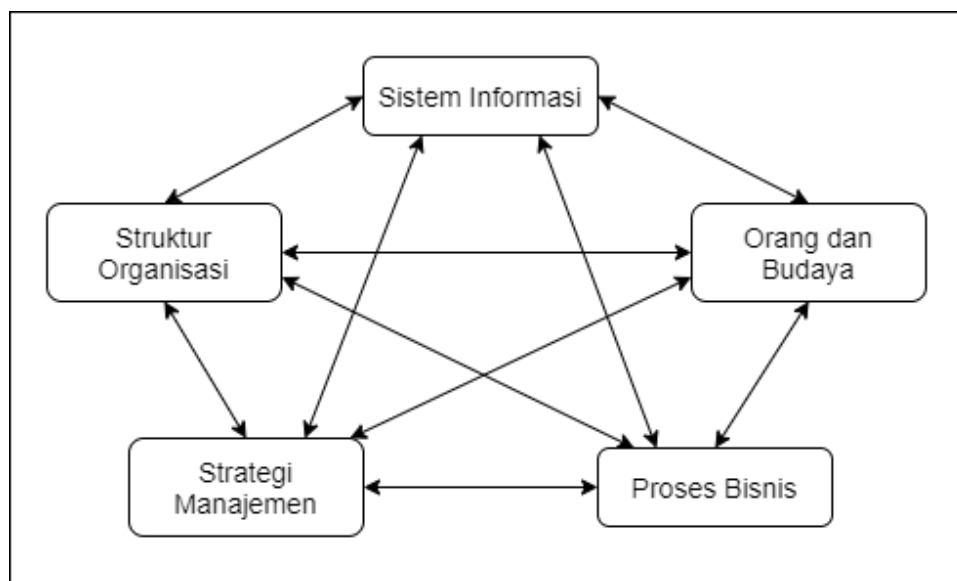
Komputer memiliki kualitas yang semakin lama semakin baik. Dengan bantuan komputer, sistem informasi mampu mengerjakan tugas-tugas yang biasa dilakukan oleh seseorang. Kegiatan ini biasa disebut dengan istilah otomasi pekerjaan. Sistem informasi juga dapat memberikan bantuan berupa informasi yang bermanfaat dalam pengambilan keputusan sehingga sangat bermanfaat bagi penggunaannya terutama para manajer atau staf suatu organisasi/perusahaan. Berikut ini peranan sistem informasi dalam suatu organisasi (Alter, 1992), yaitu [7]:

1. Berkontribusi pada pelaksanaan tugas-tugas
2. Menghubungkan tiga bagian yaitu perencanaan, pengerjaan, dan pengendalian pada sebuah subsistem

3. Mengorganisasikan subsistem-subsistem

4. Menggabungkan subsistem-subsistem

Sistem informasi merupakan salah satu komponen yang cukup penting dalam sebuah organisasi/perusahaan. Selain sistem informasi, terdapat empat komponen penting lainnya yang saling berinteraksi dan saling mempengaruhi yaitu strategi manajemen, struktur organisasi, proses bisnis, serta orang dan budaya. Berikut ini gambaran interaksi kelima komponen tersebut pada sebuah organisasi/perusahaan.



Gambar 2. 3 Gambaran interaksi lima komponen-komponen penting pada sebuah organisasi/perusahaan
(Sumber : Pengenalan Sistem Informasi Edisi Revisi [9])

2.2.4. Pengertian Akademik

Menurut Kamus Besar Bahasa Indonesia (KBBI), akademik memiliki pengertian yang sama dengan akademis. Menurut KBBI, akademis artinya sesuatu yang berhubungan dengan akademi sedangkan akademi artinya lembaga pendidikan [10]. Jadi dapat disimpulkan bahwa akademik adalah segala sesuatu yang berhubungan dengan lembaga pendidikan.

Sedangkan menurut buku Supervisi Pendidikan karya Muwahid Shulhan, akademik merupakan salah satu aspek pada supervisi terutama supervisi pendidikan [11]. Akademik atau supervisi akademik pada supervisi pendidikan sangatlah

penting karena berkenaan dengan mutu pembelajaran yang akan mempengaruhi prestasi belajar pada siswa. Menurut seorang ahli bernama glickman pada tahun 1981, supervisi akademik adalah serangkaian kegiatan untuk membantu pengembangan kemampuan guru/pengajar dalam mengelola proses pembelajaran agar dapat terlaksananya tujuan pembelajaran [11].

2.2.5. Pengertian Sistem Informasi Akademik

Sistem informasi akademik adalah sistem yang memberikan layanan informasi yang berupa data dalam hal yang berhubungan dengan akademik [12]. Dimana dalam hal ini pelayanan yang diberikan yaitu seperti : penyimpanan data untuk siswa baru, penentuan kelas, penentuan jadwal pelajaran, pembuatan jadwal mengajar, pembagian wali kelas, proses penilaian [12].

Menurut penelitian yang lain, sistem informasi akademik adalah sistem komputerisasi atau teknik pemanfaatan komputer sebagai alat bantu proses yang bertujuan untuk mengolah dan menangani data-data akademik [13]. Contohnya adalah pembuatan sebuah website yang dapat mengelola data-data akademik pada sebuah lembaga pendidikan. Pengguna dari website tersebut adalah sebagian atau seluruh elemen sekolah tergantung dari tujuan pembuatan website. Elemen sekolah diantaranya kepala sekolah, guru, staff, siswa, dan lain-lain.

2.2.6. Metode Pengembangan *Prototype*

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah metode *prototype*. *Prototype* merupakan metode yang berfokus untuk menghasilkan *prototype* sistem, hasil *prototype* sistem ini berfungsi sebagai perantara antara pengembang dan pengguna, sehingga nantinya pengguna dan pengembang dapat berinteraksi dalam proses pengembangan sebuah sistem informasi. Sebuah pembuatan *prototype* dikatakan berhasil dengan baik apabila pengembang dan pengguna mendefinisikan aturan-aturan tahap awal, yaitu pengguna dan pengembang harus satu pemahaman bahwa *prototype* dibangun mewakili kebutuhan awal [14].

Peneliti menggunakan *prototype* ini bertujuan agar pada saat pengembangan sistem bisa mengumpulkan informasi dari pengguna sehingga dapat dengan mudah untuk menggambarkan model interaksi yang akan dikembangkan.

Dalam menjalankan metode pengembangan *prototype* ini ada beberapa tahapan yang dilakukan sebagai berikut :

1. Identifikasi Kebutuhan

Tahapan pertama ini mengumpulkan kebutuhan dan melibatkan pertemuan antara pengembang sistem dengan pengguna untuk menentukan apa yang menjadi tujuan serta kebutuhan perangkat lunak atau sistem yang akan dikembangkan.

2. Membuat Prototipe

Tahapan kedua desain awal untuk pengembangan sistem Adapun untuk pembuatannya sendiri meliputi input, proses dan format output dari perangkat lunak yang akan dikembangkan

3. Menguji Prototipe

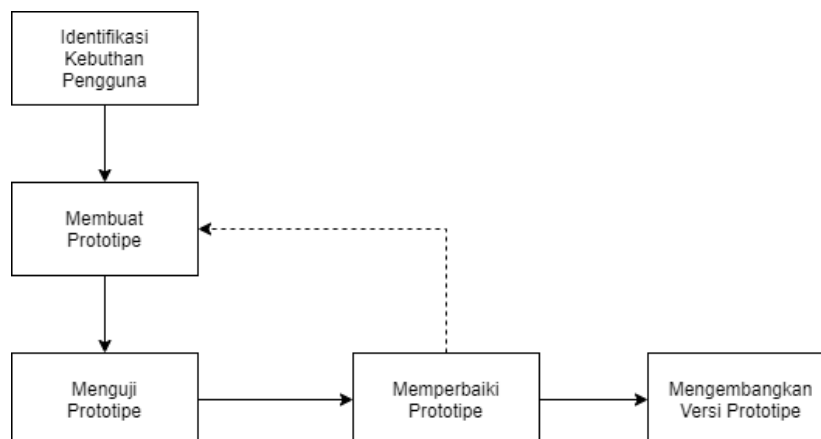
Tahapan ini sebuah protipe yang sudah dibangun akan dievaluasi oleh pengguna dan pengembang, hal ini dilakukan untuk menyesuaikan kebutuhan perangkat lunak yang dikembangkan. Pada tahap ini pengguna dan pengembang akan memahami secara detail apa yang perlu dilakukan.

4. Memperbaiki Prototipe

Setelah dilakukan pengujian dilakukan untuk perbaikan kesalahan yang didapatkan pada saat dilakukan pengujian prototipe.

5. Mengembangkan Versi Prototipe

Tahap terakhir yaitu mengembangkan versi prototipe, tujuan untuk pengembangan ke versi berikutnya agar perangkat lunak yang dikembangkan menjadi produk yang nyata dan dapat digunakan oleh pengguna dengan baik.



Gambar 2. 4 Tahapan Metode *Prototype*

2.2.7. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan bahasa yang menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan perangkat lunak yang dapat mudah dipahami oleh manusia. UML menggunakan konsep orientasi *object* yang bertujuan dapat memudahkan manusia atau pengembang memahami sistem yang akan dikembangkan [1].

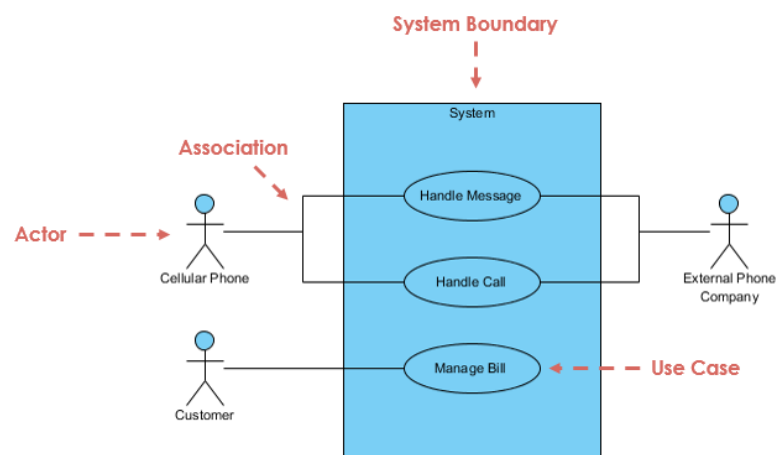
UML memiliki beberapa *diagram* yang biasanya digunakan oleh pengembang untuk melakukan pemodelan data maupun sistem, Berikut merupakan *diagram* yang ada pada UML :

1. *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah use case diagram terdiri dari aktor dan use case dimana keduanya dalam use case diagram digambarkan bagaimana aktor dan sistem saling berhubungan. Konsep dari use case ini adalah menggambarkan apa yang dilakukan oleh sistem. Contohnya adalah aktor membuat pemesanan dan sistem akan menjalankan fungsi pemesanan [1]. Pada sebuah use case ada beberapa elemen yang digunakan sebagai berikut :

- a. Aktor merupakan manusia atau sistem eksternal yang berinteraksi dengan sistem

- b. Use case merupakan fungsionalitas yang terdapat pada sebuah sistem
- c. Association merupakan sebuah garis yang menggambarkan interaksi atau hubungan aktor dengan sistem
- d. *Dependency* merupakan garis yang menggambarkan hubungan interaksi antara dua *use case*, Terdapat dua macam interaksi yaitu *include* dan *extends*. *Include* merupakan garis yang dimana menghubungkan sebuah *use case* membutuhkan *use case* lain, sedangkan *extends* merupakan sebuah garis yang dimana menggambarkan ketergantungan antara satu *use case* dengan *use case* lainnya.
- e. *Generalization* merupakan konsep perwarisan dimana aktor ataupun *use case* bisa mewariskan kepada aktor dan *use case* itu sendiri atau dari aktor ke *use case* dan sebaliknya.



Gambar 2.5 Elemen dari *Use case Diagram*
(Sumber: visual-diagram.com)

2. Use Case Scenario

Use case scenario merupakan gambaran model yang berbentuk skenario, pada use case scenario nantinya akan terlihat beberapa kemungkinan kejadian dalam menjalankan sebuah sistem tersebut. Pada proses sebuah sistem nantinya akan ada sekumpulan proses, proses inilah yang akan menjadi beberapa kemungkinan-kemungkinan yang kemudian disebut use scenario

3. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, pada *activity diagram* dijelaskan bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana sebuah *activity* berakhir. Adapun dalam *activity diagram* ada beberapa elemen yang digunakan yaitu berikut ini.

- a. *Activities* merupakan elemen yang menggambarkan aktivitas tertentu dalam sebuah sistem
- b. *Transision* merupakan elemen yang digunakan untuk menggambarkan transisi atau peralihan dari aktor ke sistem ataupun sebaliknya



Gambar 2. 6 *Gambaran Transition*
(Sumber: socs.binus.ac.id)

- c. *Initial State* merupakan elemen awal dimulainya aliran pada *activity diagram* dan pada sebuah *activity diagram* hanya terdapat satu initial state



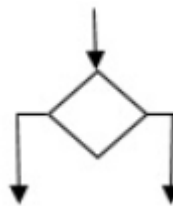
Gambar 2. 7 *Gambaran Initial State*
(Sumber: socs.binus.ac.id)

- d. *Final State* merupakan elemen bagian akhir suatu aliran kerja pada sebuah *activity diagram* dan pada sebuah *activity diagram* bisa terdapat lebih dari satu *final state*



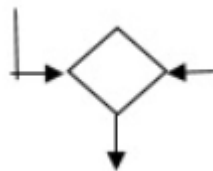
Gambar 2. 8 Gambaran *Final State*
(Sumber: socs.binus.ac.id)

- e. *Decision* merupakan elemen yang memiliki fungsi untuk menggambarkan pilihan kondisi dimana ada kemungkinan perbedaan *transition*, untuk memastikan bahwa aliran *activity* dapat mengalir ke lebih dari satu aliran.



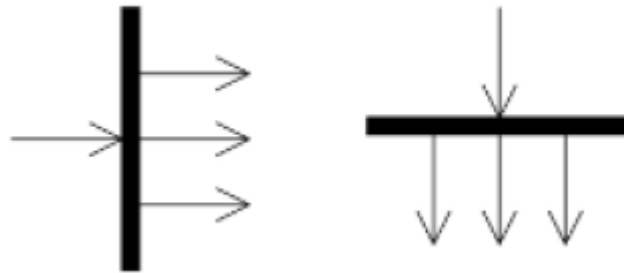
Gambar 2. 9 Gambaran *Decision*
(Sumber: socs.binus.ac.id)

- f. *Merge* merupakan elemen yang memiliki fungsi untuk menggabungkan kembali aliran *activity* yang sebelumnya telah dipisahkan oleh *decision*



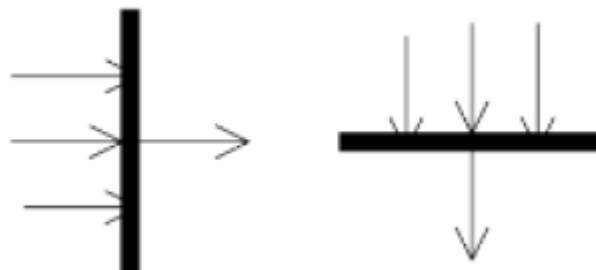
Gambar 2. 10 Gambaran *Merge*
(Sumber: socs.binus.ac.id)

- g. *Concurrency* merupakan elemen untuk melakukan percabangan proses dalam sebuah sistem. Perbedaan dengan *decision*, *decision* dilakukan untuk percabangan logika sedangkan *concurrency* dilakukan untuk percabangan proses



Gambar 2. 11 Gambaran *Concurrency*
(Sumber: socs.binus.ac.id)

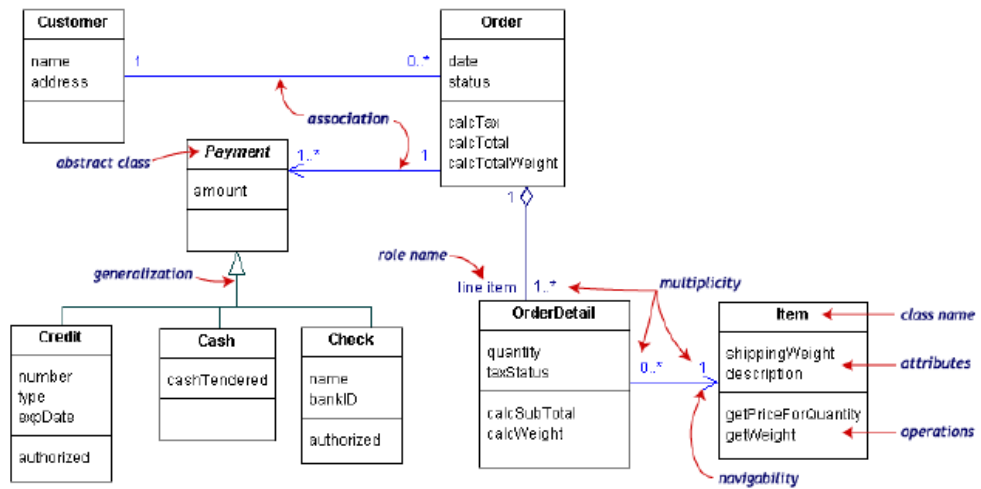
- h. *Synchronization* merupakan elemen untuk melakukan penggabungan hasil proses yang dipisahkan oleh elemen *Concurrency*



Gambar 2. 12 Gambaran *Synchronization*
(Sumber: socs.binus.ac.id)

4. *Class Diagram*

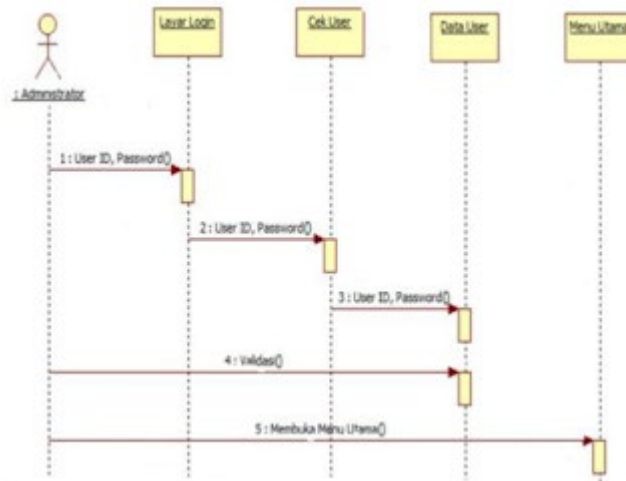
Class diagram merupakan jenis diagram struktur class yang menggambarkan struktur suatu sistem dengan menunjukkan kelas sistem, atribut, metode, dan hubungan antara objek.



Gambar 2. 13 Contoh Class Diagram
 (Sumber: Pengantar Ilmu Komputer: (UML) [13, p6])

5. Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan interaksi antara object yang ada dalam sistem maupun diluar sistem. Kegunaan dari Sequence diagram ini menunjukkan rangkaian pesan yang dikirim antara object atau sistem.



Gambar 2. 14 Contoh Sequence Diagram
 (Sumber: binus.ac.id)

2.2.8. Basis Data dan DBMS

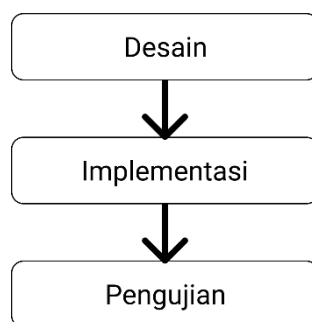
Database atau basis data merupakan kumpulan data atau informasi yang tersimpan dalam komputer secara sistematis dan terintegrasi [15]. Agar dapat

memudahkan dalam pengelolaan *database* terdapat suatu perangkat lunak yang dibutuhkan yaitu DBMS. DBMS (*DataBase Management System*) adalah sebuah perangkat lunak yang digunakan untuk membuat, memelihara, mengontrol, dan mengakses *database* secara praktis dan efisien [16] .

2.2.9. Testing Perangkat Lunak

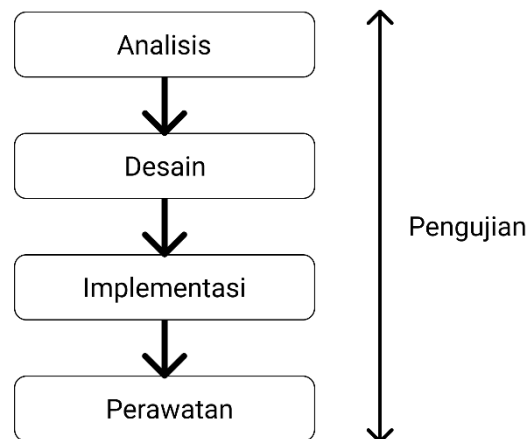
Testing perangkat lunak adalah kegiatan menjalankan perangkat lunak dengan keadaan yang dapat dikendalikan sehingga dapat diverifikasi, dicari kesalahan (*error*), dan divalidasi [17]. *Testing* perangkat lunak juga dapat didefinisikan sebagai pengukuran kualitas perangkat lunak. Perangkat lunak dikatakan berkualitas ketika terbebas dari *error* dan *bug* secara objektif, tepat waktu, dan dana sesuai dengan kebutuhan, dan dapat dirawat (*maintainable*).

Pada mulanya, pengujian merupakan salah satu tahapan dari pengembangan (*development*) perangkat lunak, yaitu setelah tahap coding. Perangkat lunak dirancang, diimplementasi, kemudian dites dan di-*debug*. Berikut ini gambaran siklus hidup testing pada masa lalu secara garis besar dapat dilihat di bawah ini.



Gambar 2. 15 Siklus Hidup Testing Dulu
(Sumber : *Testing dan Implementasi Sistem karya Hendra Jatnika [17]*)

Namun saat ini terjadi perubahan sudut pandang, pengujian yang baik dan tepat adalah pengujian yang ditempatkan pada suatu bagian dan menjadi satu kesatuan di dalam keseluruhan tahap pembangunan dan pengembangan perangkat lunak sebagaimana dapat dilihat pada gambar di bawah ini.



Gambar 2. 16 Siklus Hidup Testing Dulu
 (Sumber : Testing dan Implementasi Sistem karya Hendra Jatnika [17])

Secara umum terdapat tiga macam tingkatan pada pengujian, yaitu [17]:

1. *Unit Testing*

Pengujian pada penulisan kode-kode program dalam satuan unit terkecil secara individual.

2. *System Testing*

Pengujian pada sistem terintegrasi untuk melakukan verifikasi bahwa sistem telah sesuai dengan spesifikasi kebutuhan.

3. *Acceptance Testing*

Pengujian formal pada perangkat lunak untuk menentukan apakah dapat diterima atau tidaknya oleh pengguna akhir (*end-user*).

Pada saat tingkat *system testing*, ada yang dinamakan *test case*. *Test case* merupakan suatu tes yang dikerjakan berdasarkan pada suatu inisialisasi, *input*, keadaan, ataupun *output* yang telah ditetapkan sebelumnya.

Berdasarkan kegunaannya, *test case* terbagi menjadi 2 macam, yaitu [17]:

1. *Black Box Testing*

Digunakan untuk menguji kesesuaian suatu komponen terhadap spesifikasi kebutuhan.

2. *White Box Testing*

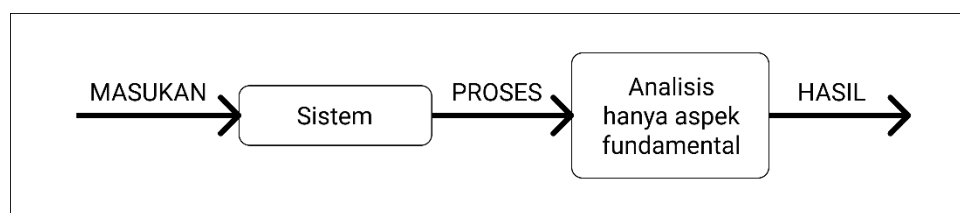
Digunakan untuk menguji kesesuaian suatu komponen terhadap desain pada sistem.

Black box testing memungkinkan perekayasa memperoleh serangkaian kondisi masukan secara keseluruhan dengan semua ketentuan fungsional pada suatu program.

Berikut ini beberapa kesalahan yang dapat ditemukan dengan black box testing [18]:

1. Fungsi-fungsi yang keliru atau hilang
2. Kekeliruan pada antar muka
3. Kekeliruan pada struktur data
4. Kekeliruan pada inialisasi dan terminasi
5. Kesesuaian pada fungsional
6. Kesensitifan pada sistem

Pada *black box testing* terdapat cukup banyak metode yang dapat digunakan. Namun secara umum, alur dari *black box testing* dapat dilihat pada gambar di bawah ini.



Gambar 2. 17 Alur Black Box Testing
(Sumber : Pengujian Perangkat Lunak dengan menggunakan Metode White Box dan Black Box karya Abdul Rouf [18])

Berikut ini beberapa metode yang dapat digunakan pada *black box testing* [18]:

1. *Limit testing*
2. *Sample testing*

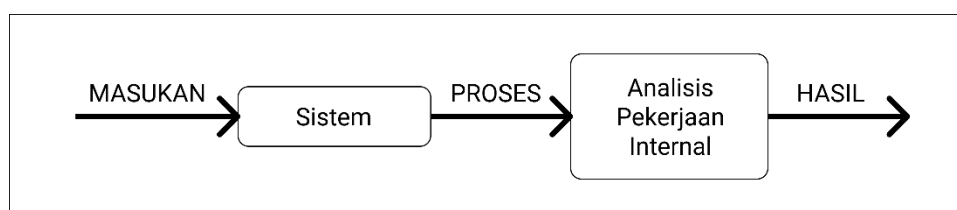
3. *Behavior testing*
4. *Robustness testing*
5. *Requirement testing*
6. *Equivalence class partitioning*

Selain *black box testing*, ada *white box testing*. Dengan *white box testing* dapat diperoleh kasus uji menggunakan struktur kontrol program.

Berikut ini beberapa kasus uji yang dapat diperoleh menggunakan *white box testing* [18]:

1. Semua jalur independen dieksekusi minimal sekali
2. Semua keputusan logikal dilakukan pengujian
3. Semua pengulangan yang sesuai batasannya diuji
4. Semua struktur data internal dilakukan pengujian

Pada *white box testing* terdapat dua metode yang paling sering digunakan yaitu *basic path testing* dan *control structure testing*. Namun secara umum alur dari *white box testing* dapat dilihat pada gambar di bawah ini.



Gambar 2. 18 Alur White Box Testing
 (Sumber : Pengujian Perangkat Lunak dengan menggunakan Metode White Box dan Black Box karya Abdul Rouf [18])

2.3. Teknologi dan Perangkat Lunak yang Digunakan

2.3.1. Pemrograman *Node.js*

Bahasa pemrograman yang digunakan adalah *javascript*. *Javascript* merupakan salah satu bahasa pemrograman web yang dapat membuat sebuah *website* menjadi lebih dinamis dan interaktif. Pada pemrograman *javascript*,

terdapat banyak teknologi yang dapat digunakan, salah satunya yaitu pemrograman *node.js*. Pemrograman *node.js* merupakan sebuah perangkat lunak yang digunakan untuk menjalankan atau mengeksekusi kode program *javascript* di luar *web browser* dan dibangun dengan *engine V8* milik Chrome [19]. Karena pada awalnya *javascript* hanya dapat berjalan di dalam sebuah *web browser*. Lalu dengan adanya *node.js*, *javascript* juga dapat dijalankan pada environment yang lebih luas yaitu sebuah sistem operasi seperti windows, mac OS, dan linux. Sebelumnya *javascript* hanya digunakan pada sisi *client*. Dengan adanya *node.js fullstack*, *javascript* dapat digunakan pada sisi *back end* dan *front end*.

Selain itu, *node.js* memiliki beberapa kelebihan lainnya sebagai berikut :

1. *Node js* bersifat *asynchronous* dimana dapat menerima permintaan selanjutnya tanpa harus menunggu permintaan sebelumnya selesai.
2. *Node js* bersifat *non-blocking* dimana dapat memproses permintaan lebih dari satu pada saat waktu yang sama sehingga terjadinya efisiensi.
3. *Node js* juga bersifat *single-threaded* dimana hanya satu thread yang digunakan sehingga penggunaan ram jauh lebih sedikit.
4. *Node.js* dibuat menggunakan *engine V8* dari Google yang memiliki kemampuan dalam mengeksekusi paling cepat dibanding *engine* lainnya.

Dikarenakan *node.js* bersifat *asynchronous*, *non-blocking*, dan *single-threaded* maka cara kerjanya berbeda dengan pemrograman lainnya yang pada umumnya bersifat sebaliknya. Bahasa pemrograman lainnya ketika program berjalan dan ketika pengambilan data dalam waktu tertentu maka *thread* berikutnya sudah harus disiapkan agar proses pengambilan data bisa berjalan dengan lancar. Sedangkan pada *node.js*, *thread* hanya akan dibuat bila ada event yang membutuhkannya saja. Hal ini tentunya akan membuat proses pengolahan data bisa dilakukan dengan lebih efisien dan lebih cepat.

2.3.2. *Cascading style sheet (CSS)*

CSS merupakan singkatan dari "*Cascading Style Sheets*". CSS adalah bahasa pemrograman yang digunakan untuk mendesain sebuah halaman *website* [20]. Seiring meningkatnya teknologi, saat ini CSS tidak hanya digunakan pada HTML dan XHTML saja tetapi sudah dapat digunakan untuk mendesain tampilan aplikasi android [20]. Ada banyak hal yang dapat dilakukan menggunakan CSS, contoh properti yang dapat diatur menggunakan CSS yaitu warna teks, jenis font, baris antar paragraf, ukuran kolom, dan masih banyak lagi.

Berikut ini beberapa alasan peneliti menggunakan CSS sebagai berikut:

1. CSS mempercepat proses desain yaitu ketika mendesain tampilan maka cukup mengetikkan satu kali fungsi CSS kemudian menggunakannya pada file HTML.
2. CSS memudahkan proses pemeliharaan yaitu ketika terjadi perubahan maka cukup mengedit satu bagian file CSS saja.
3. Dengan adanya file CSS terpisah maka satu file HTML akan mengandung lebih sedikit baris kode sehingga halaman akan lebih cepat dimuat.
4. Terakhir, CSS juga dapat membuat *website* menjadi responsive yaitu membuat konten halaman dapat disesuaikan dengan berbagai ukuran layar.

CSS menggunakan *selector* (*id* dan *class*) untuk menentukan elemen pada html yang akan dimodifikasi. Jadi dapat dikatakan bahwa HTML merupakan kerangka dari sebuah *website* dan CSS digunakan untuk mempercantik tampilan *website* tersebut. Dalam implementasinya, CSS terbagi menjadi 3 macam yaitu *external*, *internal*, dan *inline*. Tipe *internal* dan *inline* cocok digunakan ketika *website* yang dibuat masih sederhana, namun ketika *website* sudah kompleks dan memiliki jumlah halaman yang cukup banyak maka tipe *external* lebih baik untuk digunakan.

2.3.3. *Vue.js*

Pada pemrograman *javascript* terdapat sejumlah kerangka kerja (*framework*) yang dapat digunakan, salah satunya adalah *vue.js*. *Vue.js* merupakan sebuah

kerangka kerja bersifat progresif yang digunakan untuk membangun antarmuka bagi pengguna [21]. Berbeda dengan *framework* javascript lainnya, *vue.js* dirancang dari dasar agar dapat digunakan secara bertahap sesuai dengan sifatnya yaitu progresif. *Vue.js* difokuskan pada lapisan tampilan saja dan mudah untuk diintegrasikan dengan kerangka kerja lain atau proyek lain. *Vue.js* juga mendukung *single page application* ketika dikombinasikan dengan alat yang modern dan dukungan *library*.

Berikut ini alasan peneliti memilih *vue.js* sebagai kerangka kerja untuk antarmuka pengguna yaitu:

1. *Vue.js* memiliki ukuran yang relatif kecil yaitu kurang dari 5 mb.
2. *Vue.js* memiliki struktur yang sederhana sehingga mudah untuk dipahami.
3. *Vue.js* mudah untuk diintegrasikan dengan aplikasi lainnya yang dibangun dengan *javascript*.
4. *Vue.js* memiliki dokumentasi cukup lengkap dan mudah dipahami sehingga memudahkan bagi pemula untuk menggunakannya.
5. Dengan bantuan *node virtual*, *vue.js* memiliki kemampuan fleksibilitas untuk membuat *template* dalam bentuk file HTML, file *javascript*, dan file *javascript* murni.
6. *Vue.js* mudah dalam menangani blok HTML karena menggunakan arsitektur MVVM yang memungkinkan untuk komunikasi dua arah.

Saat ini *vue.js* masih terus dalam pengembangan oleh para *developer* dari beberapa negara maju dan berkembang. Versi terakhir *vue.js* yang direkomendasikan pada *website* resminya adalah *vue.js* 2.6.12. Namun, peneliti menggunakan *vue.js* versi 2.6.11 karena untuk saat ini versi tersebut masih dapat memenuhi semua kebutuhan peneliti.

2.3.4. *Material Design Framework CSS*

Material design pertama kali muncul dan diumumkan oleh google melalui konferensi Google I/O pada tahun 2014. *Material design* merupakan bahasa desain yang digunakan untuk membuat antarmuka dan gerakan yang bermakna sehingga dapat menghasilkan interaksi pengguna yang intuitif [22]. Hal dasar yang menjadi salah satu kelebihan *material design* adalah bentuk tiga dimensinya. Dengan memberikan ilusi seperti berinteraksi dengan sesuatu yang nyata.

Material design dapat digunakan pada aplikasi *mobile* dan aplikasi web. Namun, dalam mengembangkan dan mendesain aplikasi web terdapat beberapa hal yang perlu diperhatikan. Web memiliki layar yang lebih besar, kemungkinan tidak adanya layar sentuh, dan juga bervariasinya *browser* serta sistem operasi yang digunakan. Dengan adanya perbedaan tersebut, dalam mendesain aplikasi web memerlukan pendekatan yang berbeda untuk interaksi pengguna.

Pada tahap implementasi *material design*, para developer menggunakan sebuah *framework* untuk memudahkan dan mempercepat pekerjaannya. Sejauh ini, *framework* paling cepat untuk mengembangkan dan mendesain aplikasi web adalah dengan menggunakan *framework* CSS [22]. Saat ini, banyak *framework* CSS dengan konsep *material design* yang dapat digunakan seperti Materialize, MUI, Surface, dll.

2.3.5. *Database MySQL*

MySQL adalah sistem manajemen *database* yang berdsifat *open source* atau gratis [23]. Berikut ini alasan peneliti memilih menggunakan *MySQL* sebagai berikut :

1. *MySQL* dapat diintegrasikan dengan bahasa pemrograman lain sehingga dapat memudahkan proses pembangunan dari sebuah sistem.
2. *MySQL* dapat dijalankan pada komputer dengan spesifikasi *hardware* yang rendah ataupun RAM yang kecil.

3. MySQL mendukung adanya *multi-user* sehingga seluruh pengguna dapat menggunakannya pada waktu bersamaan tanpa harus menunggu pengguna lain selesai.
- 4, MySQL bersifat *open source* yaitu dapat diakses oleh siapa saja dan kapan saja. Walaupun demikian, My SQL memiliki kualitas yang cukup baik karena memiliki lisensi GPL di bawah pengelolaan Oracle.
5. MySQL memiliki struktur tabel yang fleksibel sehingga memudahkan dalam proses eksekusi terutama pada saat melakukann ALTER TABLE.
6. MySQL memiliki tipe data yang bervariasi sehingga dapat mengelompokkan data secara lebih akurat dan efektif.
7. MySQL memiliki beberapa lapisan keamanan dan juga enkripsi *password* yang tinggi sehingga keamanan datanya cukup terjamin.

2.3.6. Git

Git adalah sebuah *version control system* yang banyak digunakan oleh pengembang perangkat lunak, cara kerja git sendiri akan mencatat setiap perubahan terhadap *source code* pada perangkat lunak yang nantinya perubahan tersebut akan disimpan kedalam *internal database git*, pencatatan perubahan ini berfungsi ketika suatu saat sebuah perangkat lunak dapat dikembalikan kepada salah satu versi dari perangkat lunak tersebut [24].

2.3.7. Konsep Restful API

RESTful API adalah model implementasi dari *web service* yang memberikan layanan penghubungan antara suatu sistem untuk berinteraksi dengan sistem yang lainnya berbagi data. *Restful API* memungkinkan komunikasi antara *client* dan *server* menggunakan metode *HTTP* dan me-*return response* data berupa JSON (*javascript object notation*) yang nanti nya diolah dari sisi *client* [25].