



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Visualisasi

Menurut Khan (2011), visualisasi merupakan sebuah metode yang digunakan untuk mendeskripsikan sebuah gambaran ide ataupun data, sehingga dapat membantu menangkap dan menganalisa suatu permasalahan, melalui grafik ataupun gambar dengan menggunakan media komputer secara interaktif.

Secara umum ada beberapa aktifitas yang harus dilakukan sehingga data tersebut bisa divisualisasikan. Aktifitas-aktifitas tersebut terdiri dari:

- a. **Mapping** : aktifitas dimana peneliti melakukan analisa data dan menentukan bagaimana visualisasi akan dilakukan. Semakin baik visualisasi dilakukan, maka akan semakin akurat tingkat visualisasi.
- b. **Selection** : aktifitas memilih data yang dipakai dalam lingkup permasalahan, pemilihan data harus dilakukan dengan hati-hati sehingga data visualisasi bisa menjalankan fungsinya secara maksimal.
- c. **Presentation** : aktifitas dimana perancang visualisasi melakukan penyajian informasi menggunakan data yang tersedia, sehingga dapat menjalankan fungsinya secara efektif
- d. **Interactivity**: aktifitas dimana perancang visualisasi memberikan kesempatan kepada pengguna, dalam mengeksplorasi data yang telah divisualisasi oleh perancang. Pengguna visualisasi dapat menelusuri

visualisasi yang dibuat dan melakukan interaksi dengan data sesuai dengan kebutuhan.

- e. **Human Factor** adalah faktor manusia yang perlu dipertimbangkan dalam memvisualisasikan data. Faktor yang perlu dipertimbangkan tersebut adalah *usability* dan *Accessibilty* pengguna.
- f. **Evaluation** adalah aktifitas dimana pengguna melakukan penilaian terhadap visualisasi sesuai dengan lingkup permasalahan dan tingkat kemudahan pengguna.

Terdapat beberapa jenis visualisasi yang dilakukan dalam membantu berbagai macam bidang antara lain: visualisasi sains, visualisasi informasi, visualisasi konsep, visualisasi strategi, visualisasi data, visualisasi metaphore, dan visualisasi *compound*.

Dari beberapa jenis visualisasi di atas, dua jenis metode yang dipakai di dalam penelitian ini:

A. Visualisasi Data

Menurut Khan (2011), visualisasi data merupakan metode visualisasi yang menggambarkan data secara sistematis sehingga dapat disajikan kepada pengguna, dan membantu pengguna dalam melakukan analisa data. Ada banyak jenis-jenis visualisasi data yang selama ini kita kenal contohnya:

a. Grafik pie (donat, pie):

Grafik lingkaran yang terbagi menjadi beberapa bagian, dimana setiap bagian mendefinisikan masing-masing data.

b. Grafik batang:

Menvisualisasikan data dalam bentuk batang sesuai dengan indikator data yang ditentukan.

c. Grafik *bubble*:

Grafik yang didefinisikan dengan 3 parameter angka yang digunakan untuk menunjukkan keberadaan data didalam grafik

d. Grafik *Scatter plot*:

Grafik yang menunjukkan sebuah set data di dalam koordinat kartesian dengan maksud mendefinisikan hubungan antar dua buah variable, variabel bebas menunjukkan jarak mendatar dan variabel terikat menunjukkan jarak tegaklurus

e. Histogram:

Grafik yang digunakan untuk statistik dan analisis data. Biasanya digunakan untuk mendeskripsikan distribusi data

f. Tabel:

Cara mempresentasikan data yang paling banyak digunakan, terdiri dari baris dan kolom yang digunakan untuk menunjukkan hubungan antara 2 set data.

g. Grafik garis:

Grafik yang digunakan untuk menunjukkan data yang terhubung antara satu titik dengan titik lainnya menjadi sebuah garis

h. *Multiple chart:*

Gabungan dari beberapa grafik yang digunakan untuk mendeskripsikan sebuah data yang sifatnya besar (tidak terpecah dalam 1 atau dua parameter saja).

B. Visualisasi Informasi

Menurut Khan (2011), visualisasi informasi merupakan domain penelitian yang berkonsentrasi pada penggunaan visualisasi sebagai metode yang membantu seseorang dalam mengerti data sebagai informasi yang disediakan untuk disajikan.

Berikut ini merupakan contoh-contoh yang digunakan untuk visualisasi informasi:

a. *Parallel coordinat:*

Visualisasi yang digunakan untuk melambangkan data yang bersifat individual yang ada pada berbagai dimensi data. Setiap data yang berhubungan akan ditampilkan sebagai rangkaian garis.

b. *Tree Map:*

Visualisasi yang digunakan untuk menampilkan sebuah informasi yang bersifat hirarki yang ditampilkan dengan sebuah kotak yang berlayer.

c. *Diagram Relasi objek:*

Digunakan untuk mendeskripsikan hubungan antara data dalam membangun sebuah software, bisa disebut dengan desain *database* yang akan digunakan pada pembangunan sistem.

d. *Time Line:*

Digunakan untuk mendeskripsikan kronologi kejadian yang terjadi dalam jangka waktu tertentu. Timeline mungkin secara tabular atau disebut dengan rangkaian waktu tertentu seperti tahunan, bulanan, harian, dan lainnya.

e. *Flow chart:*

Digunakan untuk mendeskripsikan langkah yang terlibat dalam sebuah proses. Selain menggambarkan langkah-langkah proses *flow chart* juga menggambarkan aliran data secara urut.

f. *Data flow Diagram:*

Digunakan untuk menggambarkan bagaimana pemrosesan data di dalam sistem.

g. *Venn Diagram:*

Digunakan untuk menjelaskan relasi antara dua atau lebih data, yang bisa disebut juga dengan set diagram.

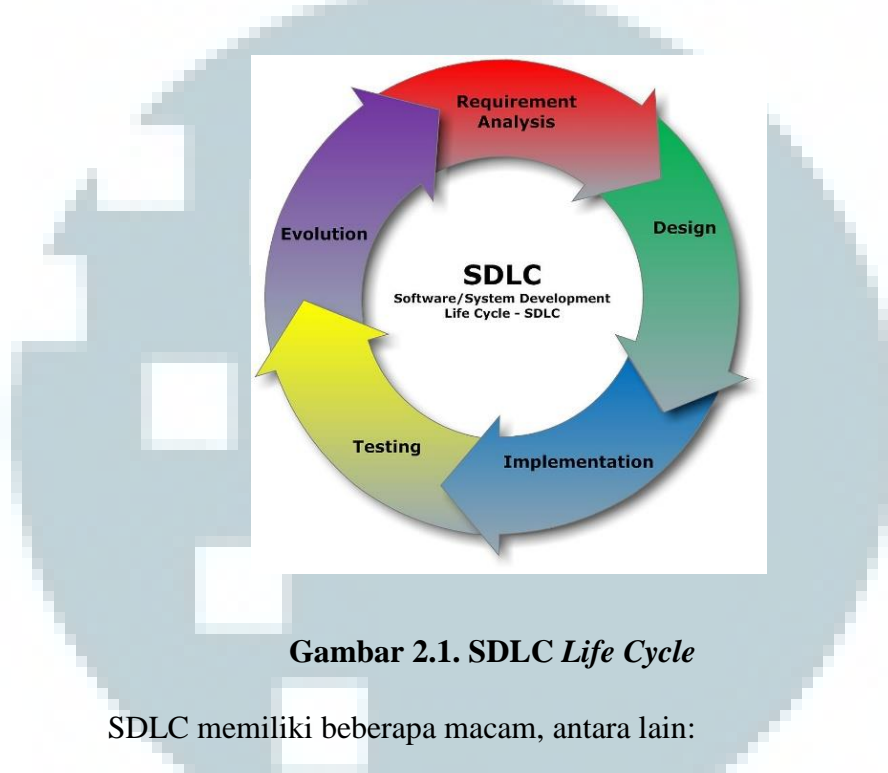
h. *Semantic Network:*

Merupakan representasi secara grafik tentang relasi secara logika antara beberapa konsep secara sekaligus.

2.2. SDLC

Menurut Rastogi (2015), SDLC (*System Development Life Cycle*) merupakan metode yang digunakan dalam pembangunan sistem agar proses pembangunan sistem dapat direncanakan, diatur, dan dikontrol oleh pengguna sehingga proses dapat berjalan dan terukur.

Siklus SDLC meliputi perencanaan dan visualisasi, analisis kebutuhan, pemeliharaan sistem dan desain, koding/pengembangan, sistem dokumentasi, *testing*, peluncuran dan pemeliharaan sistem.



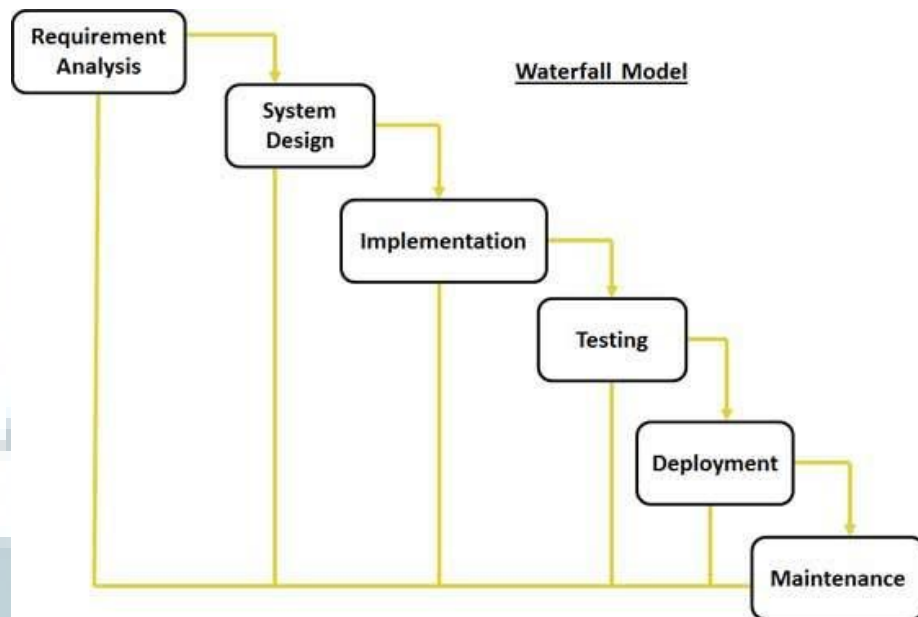
Gambar 2.1. SDLC Life Cycle

SDLC memiliki beberapa macam, antara lain:

2.2.1. SDLC Waterfall

Menurut Rastogi (2015), model ini merupakan model yang paling umum digunakan dalam pembangunan sistem berskala besar. Metode SDLC *waterfall* banyak digunakan oleh pembangun sistem yang sudah dapat mengetahui kebutuhan sistem secara jelas.

Metode SDLC *Waterfall* dilakukan secara bertahap sesuai dengan urutannya.



Gambar 2.2. SDLC Waterfall

SDLC *Waterfall* mempunyai tahapan sebagai berikut:

- a. ***Requirement Gathering and Analysis*** : mengidentifikasi kebutuhan pengguna berdasarkan wawancara untuk kemudian mendokumentasikan kebutuhan tersebut.
- b. ***System Design***: dari kebutuhan yang sudah terdokumentasi maka dibuat analisis sistem untuk membangun sistem yang ditujukan sesuai dengan kebutuhan pengguna.
- c. ***Implementation***: dengan adanya input dari sistem desain maka akan dibangun unit-unit sistem yang lebih kecil dimana pada fase selanjutnya akan digabungkan menjadi satu. Setiap unit harus sudah teruji fungsinya.

- d. **Integration and Testing:** unit yang sudah dibangun diintegrasikan, untuk kemudian diuji keberhasilan berjalannya fungsi yang digunakan dalam sistem.
- e. **System Deployment:** mengimplementasikan sistem yang telah terintegrasi dan sudah dites untuk dipakai oleh pengguna. Sistem harus sudah di-*testing* dan berjalan sesuai dengan fungsi
- f. **Maintenance:** pemeliharaan terhadap sistem yang sudah diimplementasikan, apabila masih terdapat ketidaksesuaian maka akan diperbaiki dan jika ada modul yang ingin ditambahkan maka sistem akan ditambahkan.

Pada tahap awal perencanaan sistem selain melakukan analisis terhadap kebutuhan pengguna, dilakukan juga perencanaan terhadap biaya dan analisis terhadap risiko yang mungkin dideteksi. Setelah perencanaan dibuat maka pembangun melakukan pembangunan sistem sesuai dengan analisis yang telah dilakukan. Setelah sistem selesai dibuat, sistem diimplementasikan untuk kemudian dilakukan pengujian. Bila Hasil pengujian tersebut sukses maka untuk selanjutnya dilakukan pemeliharaan.

Tahap tersebut harus dilakukan sesuai dengan urutan dan pada setiap tahap yang ada pada siklus SDLC, perlu diberikan kontrol yang ketat. Setiap hasil pada urutan pengerjaan dalam SDLC diperiksa secara formal dan disetujui oleh pengguna dan manajemen IT.

2.2.2. SDLC Agile

Menurut Rastogi (2015), SDLC *Agile* merupakan model SDLC yang digunakan dengan menyampaikan sebuah sistem yang fungsional untuk memenuhi kebutuhan pengguna. Sistem yang fungsional ini dikembangkan secara terus menerus sesuai dengan kebutuhan pengguna yang mungkin berubah.

Proses SDLC *Agile* meliputi proses berikut:

- a. *Kicking off meeting*
- b. Kebutuhan yang diketahui dibuat sebagai prioritas,
- c. Perkiraan tingkat kesulitan sistem sesuai dengan masing-masing kebutuhan.
- d. Mendesain sistem dengan menggunakan diagram sehingga mudah dimengerti,
- e. Pendekatan *Test Driven Development* bila diperlukan dengan membuat tes tertulis tentang kebutuhan pengguna kemudian dibuatkan kodenya.
- f. Pengembangan dilakukan biasanya per kelompok, sehingga antar kelompok internal dan eksternal terdapat interaksi.
- g. Kode dites berkelanjutan dimana memungkinkan penggunaan *software* dan *resources*.
- h. Penilaian terhadap sistem dan *refactoring* akan merubah sedikit kode yang perlu diperbaharui atau ditambah per fungsi bagian sistem.

2.2.3. SDLC Agile VS SDLC Waterfall

Berdasarkan teori di atas maka berikut perbandingan antara kedua metode pembangunan.

Tabel 2.1. Perbandingan SDLC Agile dan SDLC Waterfall

Parameter	SDLC Waterfall	SDLC Agile
<i>Conceptual design</i>	<i>Waterfall</i> model dilakukan secara sequensial sehingga harus fase per fase diselesaikan.	<i>Agile</i> model fokus di kecepatan dan kemampuan beradaptasi sistem.
Efisiensi	Tidak efisien karena dilakukan bertahap sesuai dengan fase-fase yang telah ditentukan, Sehingga jika ingin melakukan perubahan memakan waktu	Jika ingin melakukan pengembangan lebih jauh dapat dilakukan secara cepat, karena SDLC <i>Agile</i> bersifat fleksibel sehingga bisa sifatnya cepat.
Kecocokan	<i>Waterfall</i> digunakan pada saat kebutuhan pengguna jelas.	SDLC <i>Agile</i> bisa beradaptasi dalam segala jenis pengembangan software.
<i>Conceptual design</i>	<i>Waterfall</i> model menggunakan sequensial sehingga harus fase per fase diselesaikan	model <i>Agile</i> berfokus di kecepatan dan kemampuan beradaptasi sistem

Scrum merupakan sebuah *framework* untuk model pembangunan sistem *Agile* dalam melakukan manajemen, peningkatan, pemeliharaan terhadap sistem atau produk *prototype* (scrumalliance.org, 2011). Di dalam menggunakan scrum filosofi terhadap pembangunan sistem didefinisikan sebagai sebuah set aktifitas yang terintegrasi dengan tingkat fleksibilitas, *Responsive*, reliabilitas yang tinggi.

Scrum *Agile* memiliki karakteristik sebagai berikut:

- a. Tahap pertama dan terakhir terdiri dari proses yang didefinisikan
- b. Sprint merupakan proses empiris sehingga tidak dapat didefinisikan dan dikontrol
- c. Sprint bersifat non linier dan fleksibel
- d. Proyek dimulai dari pembukaan hingga penutupan per fase *sprint*.

Didalam tim scrum terdiri dari 2 team yaitu tim manajemen dan *development*, dimana masing-masing mempunyai sebuah orang yang bertanggung jawab atas sistem. Dari manajemen, orang yang bertanggung jawab disebut pemilik sistem, sementara dari pihak pembangun disebut *scrum master* yang bertugas menjamin bahwa proses pembangunan berjalan sesuai dengan jadwal.

Keuntungan dari scrum *Agile* ialah desain yang fleksibel dimana setiap fase desain mempunyai kontrol terhadap perencanaan yang dibuat. Scrum juga bersifat bebas terutama bagi pihak pembangun sesuai lingkungan yang berubah. Dengan tim yang kecil maka pembangun bisa berbagi pengetahuan tentang proses pembangunan, sehingga merupakan lingkungan yang bagus untuk belajar.

2.3. Web Event Driven Calendar

Menurut Benerson Richard (2001), sistem penjadwalan mengijinkan pengguna untuk meminta *event* kepada server sebagai pengingat. Untuk dapat disebut sebagai *web event driven calendar* maka sistem harus memenuhi beberapa kriteria:

- a. Pengingat akan secara otomatis dimutakhirkan sesuai dengan permintaan pengguna.

- b. Peningat bisa memberikan notifikasi tentang *event* yang diminta pengguna melalui berbagai media.
- c. Peningat dapat secara otomatis memperbaharui *event calendar* sesuai dengan perubahan data *event*.

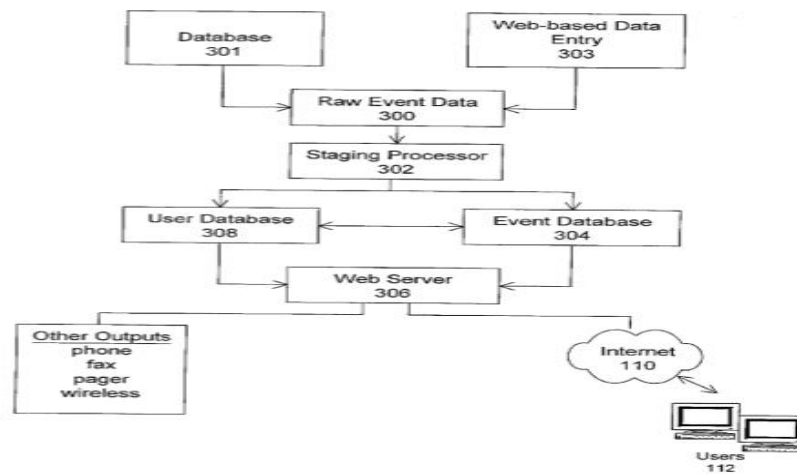
Dalam praktiknya *web event driven calendar* sudah banyak digunakan oleh berbagai *website* dan perusahaan besar lainnya, seperti Google. Pada dasarnya sistem ini bekerja dengan proses yang akan dijelaskan sebagai berikut:

Pengguna membuat *event* yang nantinya ditampilkan ke dalam kalender sebagai *event object*, sesuai dengan tanggal yang telah ditentukan oleh pengguna.

Setelah pengguna mendefinisikan *event*, maka akan menjadi data mentah yang diproses menjadi sebuah format. Format data mentah, menyesuaikan dengan tabel yang ada di dalam *database* yang berisi informasi *event* pengguna. *Event* tersebut ditampilkan ke dalam kalender sebagai daftar event yang dimiliki pengguna sesuai dengan event yang dimasukkan.

Selain ditampilkan kepada pengguna, sebagai peningat biasanya akan dikirimkan *email* ataupun jenis peningat lainnya baik secara manual maupun otomatis kepada pengguna yang bersangkutan.

Proses ini dilakukan *real time* dan terkoneksi antar pengguna sehingga perlu dilakukan melalui internet.



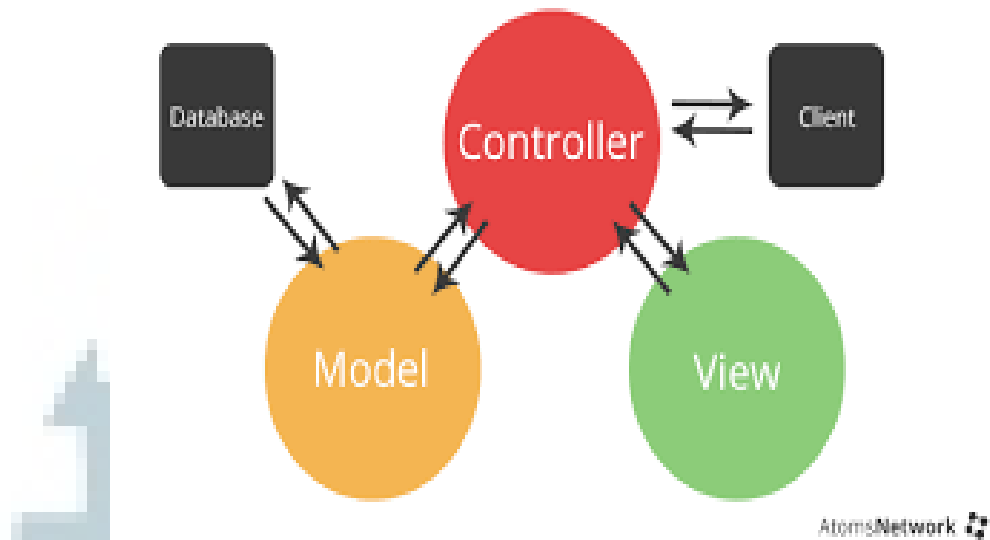
Gambar 2.3. Web Event Driven Calendar Process

2.4. MVC-R model

Menurut Iqbal, dkk (2014), MVC-R merupakan model pengembangan aplikasi yang memisahkan antara tampilan dengan logika yang berjalan mendukung aplikasi. MVC-R model, sudah digunakan di banyak *framework*.

Dalam tugas akhir ini pemecahan solusi yang dilakukan, menggunakan *framework* Laravel dan angular JS yang menerapkan konsep MVC-R

U
M
M
N



Gambar 2.4. MVC Architecture

Sesuai dengan namanya MVC-R model terdiri dar 4 bagian:

2.4.1. Model

Merupakan bagian yang mendefinisikan objek data yang digunakan. Selain itu juga dapat digunakan untuk memutakhirkan data yang ada ketika dimodifikasi.

2.4.2. View

Merupakan bagian yang menampilkan data kepada pengguna sehingga bisa dilihat. Selain menampilkan data, *view* juga merupakan tempat melakukan interaksi dengan objek data yang akan dimanipulasi oleh pengguna sendiri

2.4.3. Controller

Merupakan bagian yang berlaku di antara *model dan view*. *Controller* mengubah aliran data menjadi objek model dan memperbaharui view ketika data berubah, sehingga *view* dan *model* dapat dipisahkan.

2.4.4. Router

Merupakan bagian dari MVC-R sebagai sarana komunikasi antara *back-end* dan *front-end*, yang merupakan *gateway* komunikasi data. Data API tersebut dipanggil melalui alamat URL (*Uniform Resources Locator*) yang langsung menuju kepada data tempat pemanggilan untuk diolah dan dipanggil ke aplikasi klien untuk ditampilkan.

2.5. REST API

Menurut Asnika, dkk (2014), REST merupakan prinsip arsitektur sistem yang terdiri dari batasan yang diaplikasikan di dalam komponen seperti *connector* dan elemen data, yang terdistribusi melalui *hypermedia system*.

Konsep REST dijelaskan sebagai prinsip arsitektur dimana data bisa di kirim dan dikomunikasikan melalui *interface* yang terstandar seperti HTTP. Setiap pengguna bisa mengakses URI (*Uniform Resource Identifier*) dalam bentuk URL sebagai representasi metode yang akan dipakai untuk berinteraksi dengan data. Terdapat beberapa metode pengenalan seperti GET, POST, PUT, DELETE, PATCH.

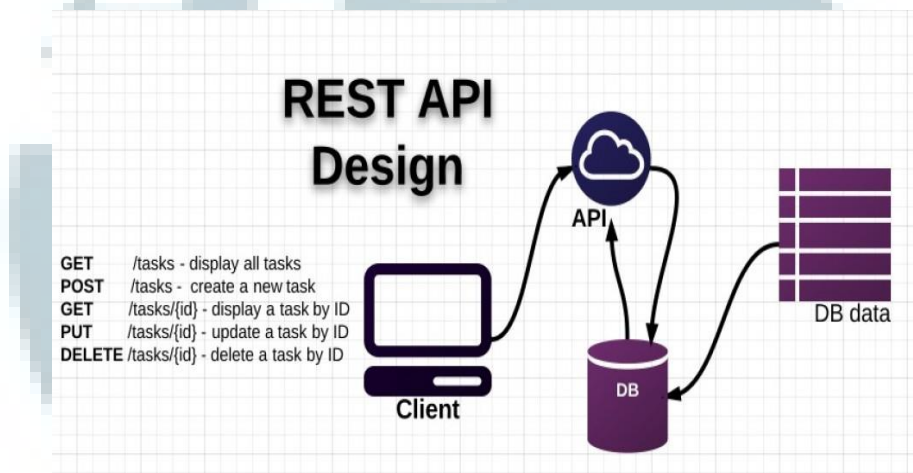
Metode pengenalan tersebut akan bekerja sebagai pengenalan yang menjelaskan operasi yang akan dilakukan. Berikut penjelasan beberapa metode pengenalan tersebut:

- a. **GET**: merupakan metode *http.get* yang digunakan dalam komunikasi yang bertujuan untuk merequest data dari server
- b. **POST**: merupakan metode http yang digunakan untuk mengirim data antara *engine* sehingga ada modifikasi data ataupun penambahan data melalui request *http.post*
- c. **DELETE**: metode http yang digunakan untuk menghapus data yang ada melalui *request http.delete*
- d. **PATCH**: metode http yang digunakan untuk memodifikasi sebuah data yang digunakan melalui *request http.patch*.

Setiap metode diatas yang didefinisikan dalam suatu URL dikenal dengan istilah API, yang digunakan oleh pengguna sebagai pengenalan metode untuk berinteraksi dengan data. Keuntungan memakai konsep REST ialah sebagai berikut:

- REST API karena bersifat terdistribusi maka sifatnya lebih cepat
- REST API bersifat ringan dan bisa digunakan oleh banyak *tools* yang dipakai dalam pembangunan arsitektur.
- REST API bersifat *stateless*, artinya mudah untuk di uji coba. Hanya dengan mengulang server dan melihat apakah interface dapat berjalan dan bertahan.

- REST API menyediakan infrastruktur *caching* yang baik dengan menggunakan HTTP GET, sehingga dengan meningkatkan infrastruktur yang baik akan meningkatkan performa server.
- REST API : Mudah diintegrasikan oleh banyak alat sehingga tidak terpusat oleh satu alat saja.



Gambar 2.5. REST API Architecture

2.6. Testing System

Menurut Sharma (2014), *testing system* merupakan percobaan dan pemeriksaan yang dilakukan secara sistematis bertujuan untuk menjamin kualitas software. *Testing system* tidak hanya menjamin kualitas saja selain itu juga menjamin agar produk sistem berjalan lancar sesuai dengan kriteria tertentu.

Ada 2 jenis *testing* di dalam *software testing* yaitu *automated* dan *manual testing*.

A. Automated Testing

Automated Testing merupakan teknik *testing* yang dimana *engineer* membangun skrip untuk melakukan pengujian. *Test case* dilaksanakan oleh komputer dengan intervensi yang minimum dari manusia.

Benefit *automated testing*:

- a. **Biaya yang efektif:** *testing* otomatis mengurangi biaya *engineer* dalam melakukan *testing*.
- b. **Penggunaan skrip yang berulang:** *testing* yang sama bisa dilakukan berulang dengan skrip yang sama.
- c. **Bisa didaur ulang:** kode skrip yang telah dipakai bisa dipakai kembali untuk *testing* yang berbeda.
- d. **Bisa di program:** *engineer* bisa memprogram secara bebas test sesuai dengan kebutuhan.
- e. **Komprehensif:** *engineer* bisa membangun test unit yang mencakup seluruh test unit.
- f. **Bisa diandalkan:** *automated testing* yang dilakukan bisa lebih dipercaya daripada *manual testing*.
- g. **Test Coverage :** *testing* yang dilakukan bisa mencakup area yang luas

B. Manual Testing

Testing yang dilakukan dengan mempersiapkan *test case* secara manual untuk mendeteksi kekurangan *software*. Tidak semua orang dalam *testing* bisa ikut serta untuk mencakup satu *test case*.

Permasalahan *manual testing*:

- a. Menghabiskan waktu dan tenaga.
- b. Investasinya besar dalam melakukan *testing*.
- c. Tidak reliabel.
- d. Tidak bisa di program sehingga tida leluasa.

2.7. UML (*Unified Modeling Language*)

UML merupakan bahasa yang dipakai dalam mendesain sistem untuk menjelaskan, memvisualisasikan, membangun, dan mendokumentasikan sistem yang akan dibangun (UML.org, 2017).

UML bisa disebut juga merupakan pemodelan sistem secara simple yang menggambarkan keseluruhan sistem yang kompleks sehingga lebih mudah dimengerti. Konsep pemodelan UML dibuat berdasarkan *Object Oriented System*.

Dalam melakukan pembuat UML terdapat 3 model:

- a. *Structural Modeling*: merupakan model yang membahas tentang sisi statis sistem.
- b. *Behavioural Modeling*: merupakan model yang membahas tentang interaksi yang terjadi dalam sistem.
- c. *Architectural Modeling*: merupakan model yang menggambarkan tentang rangkuman sistem.

Dalam tugas akhir ini dalam melakukan desain pengembangan visualisasi kalender SPMI, akan dilakukan desain sistem terhadap model *Strucural* dan *Behavioural Modeling* yang menggambarkan sistem secara keseluruhan.

2.8. Wawancara

Menurut Lukman (2013), wawancara adalah situasi dimana pewawancara dan responden berhadapan. Wawancara dilakukan guna menggali informasi yang dibutuhkan dengan kemungkinan salah yang minimum dan efisiensi yang maksimum.

Terdapat 2 jenis yaitu wawancara informal dan wawancara formal. Wawancara formal merupakan wawancara terstruktur yang menggunakan sebuah prosedur dalam menggali informasi terhadap responden, dengan rangkaian pertanyaan yang telah dipersiapkan terlebih dahulu. Wawancara informal merupakan wawancara yang dimana tidak terstruktur dan tidak direncanakan dulu informasi yang digali. Pewawancara dapat bertanya dengan kuasa penuh terhadap informasi yang ingin digali.

Wawancara formal mempunyai kelebihan dalam hal kesesuaian informasi yang lebih tinggi dibanding dengan wawancara informal, namun memakan waktu lama dan memakan biaya yang besar karena pewawancara perlu menyusun wawancaranya secara terstruktur. Wawancara informal memakan waktu dan biaya yang sedikit walau kesesuaian informasi tidak setinggi wawancara formal, sehingga wawancara informal lebih banyak digunakan. Selain dengan biaya yang sedikit dan waktu yang singkat, wawancara informal dapat menggali lebih dalam sesuai dengan kebutuhan pewawancara.