

BAB 2

LANDASAN TEORI

2.1 Tinjauan Sekolah

2.1.1 Sejarah Sekolah

SLB Rafaha Arjasari berdiri sejak tahun 2014 dan berkedudukan di Kabupaten Bandung, yang tepatnya beralamat di Jln. Kenangasari, Arjasari. SLB Rafaha Arjasari adalah salah satu sekolah yang menyelenggarakan pendidikan khusus bagi anak berkebutuhan khusus atau ABK. Pendirian sekolah ini didasari oleh keprihatinan mengenai banyaknya anak-anak berkebutuhan khusus di Kecamatan Arjasari yang tidak mendapatkan pendidikan. Pada awalnya pembelajaran dilakukan dengan meminjam ruangan dari salah satu SD di Arjasari, baru pada tahun 2015 SLB Rafaha memiliki bangunan tersendiri dan mendapat legalitas sekolah dengan Ijin Pendirian Nomor AHU- 02189.50.10.2014 dan Ijin Operasional : 421.9/Kep.10/I/SLB-BPMPT/X/2015.

2.1.2 Visi dan Misi Sekolah

Visi :

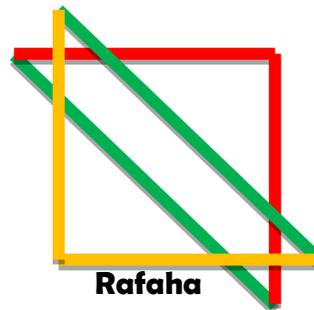
Memberikan pelayanan kepada siswa berkebutuhan khusus agar bisa mandiri, berkemampuan optimal dan berakhlak mulia. Terbentuknya lulusan yang mandiri, berpengetahuan, trampil dan berakhlak mulia sampai 2019.

Misi :

1. Menyelenggarakan pendidikan keterampilan untuk beradaptasi dengan lingkungan.
2. Mengoptimalkan pembelajaran keagamaan.
3. Melatih anak guna memiliki kecakapan hidup.
4. Menyediakan media dan sarana belajar.
5. Mempersiapkan siswa memiliki keterampilan dan kemandirian.

2.1.3 Logo Sekolah

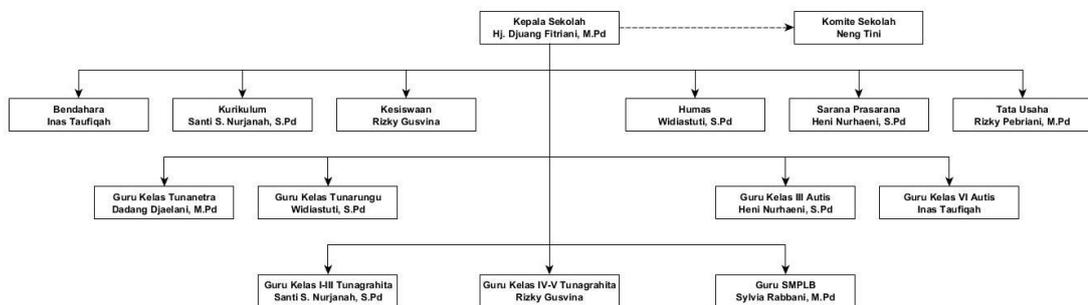
Logo sekolah SLB Rafaha Arjasari adalah sebagai berikut :



Gambar 2.1 Logo Sekolah SLB Rafaha Arjasari

2.1.4 Struktur Organisasi

Struktur organisasi yang ada di SLB Rafaha Arjasari dapat digambarkan sebagai berikut :



Gambar 2.2 Struktur Organisasi SLB Rafaha Arjasari

2.2 Kajian Teori

Pada sub bab ini akan dijelaskan mengenai teori teori terkait dengan penelitian yang diangkat dari beberapa sumber, yaitu :

1. Anak Berkebutuhan Khusus
2. Asesmen
3. *Progressive Web App*
4. *React.js*

5. API
6. *CodeIgniter*
7. *Unified Modelling Language* (UML)

2.2.1 Anak Berkebutuhan Khusus

Kementerian Pemberdayaan Perempuan dan Perlindungan Anak Republik Indonesia 2013, menjelaskan bahwa anak berkebutuhan khusus adalah: “Anak yang mengalami keterbatasan atau keluarbiasaan, baik fisik, mental-intelektual, sosial, maupun emosional, yang berpengaruh secara signifikan dalam proses pertumbuhan atau perkembangannya dibandingkan dengan anak-anak lain yang seusia dengannya”.

Anak berkebutuhan khusus adalah sebutan bagi mereka yang mengalami keadaan diri berbeda dari anak-anak pada umumnya. Beragam istilah untuk anak berkebutuhan khusus (*special needs*) ini diantaranya adalah *exceptional* (berbeda dari orang pada umumnya), *impairment* (rusak atau cacat atau sakit, lebih pada makna medis), *handicap* (tidak bisa mengakses lingkungan), dan *disability* (tidak ada atau kurangnya fungsi). Beberapa ahli juga menyebut anak berkebutuhan khusus adalah mereka yang memiliki disabilitas intelektual dan perkembangan, disabilitas fisik, disabilitas sensoris (indera), hambatan/masalah perilaku, kesulitan belajar, serta cerdas dan bakat istimewa (Blackhurst, A. E., & Berdine, W. H., 1981: 7-8). [9]

Menurut Abdullah [10] anak yang dikategorikan berkebutuhan dalam aspek fisik meliputi kelainan dalam indra penglihatan (tunanetra) kelainan indra pendengaran (tuna rungu) kelainan kemampuan berbicara (tuna wicara) dan kelainan fungsi anggota tubuh (tuna daksa). Anak yang memiliki kebutuhan dalam aspek mental meliputi anak yang memiliki kemampuan mental lebih (super normal) yang dikenal sebagai anak berbakat atau anak unggul dan yang memiliki kemampuan mental sangat rendah (abnormal) yang dikenal sebagai tuna grahita. Anak yang memiliki kelainan dalam aspek sosial adalah anak yang memiliki

kesulitan dalam menyesuaikan perilakunya terhadap lingkungan sekitarnya. Anak yang termasuk dalam kelompok ini dikenal dengan sebutan tunalaras [10].

Maka dari beberapa pengertian yang telah dipaparkan diatas, dapat diimpulkan bahwa anak berkebutuhsn khusus adalah anak yang memiliki kebutuhan khusus dikarenakan memiliki hambatan dalam fisik atau kecerdasan dan perkembangan, hambatan dalam perilaku, kesulitan belajar serta memiliki bakat istimewa dengan intelegensi yang tinggi.

2.2.2 Asesmen

Istilah asesmen dalam pedidikan khusus memiliki makna yang berbeda dengan asesmen yang digunakan secara umum dalam dunia pendidikan. Pada umumnya orang sering menterjemahkan istilah asesmen sebagai penilaian, padahal sesungguhnya terjemahan itu tidak cocok, sebab asesmen dalam pendidikan khusus memiliki pengertian yang khas. Sejalan dengan hal ini, Sari mengartikan asesmen sebagai proses mengumpulkan, menganalisis dan menginterpretasikan data atau informasi tentang peserta didik dan lingkungannya” [2]

Asesmen dalam pendidikan khusus dapat dikelompokkan menjadi dua kategori yaitu: 1) asesmen berazaskan kurikulum (asesmen akademik), dan 2) asesmen berazaskan perkembangan (asesmen non-akademik). Sejalan dengan ruang lingkup asesmen, asesmen diartikan sebagai suatu proses untuk mengambil keputusan dengan menggunakan informasi yang diperoleh melalui pengukuran hasil belajar yaitu menggunakan instrumen test maupun non test. [11]

Sementara itu Hartati [12] menyebutkan bahwa asesmen perkembangan anak merupakan suatu proses yang dilakukan secara sistematis dalam usaha untuk memperoleh informasi tentang kemampuan dan perkembangan anak. Hasil asesmen akan menjadi laporan atau informasi bagi guru dan orangtua dalam merancang program dan kebutuhan yang sesuai dengan pencapaian. [12]

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa Asesmen adalah upaya untuk mengetahui kemampuan-kemampuan yang dimiliki, hambatan

atau kesulitan yang dialami, mengetahui latar belakang mengapa hambatan atau kesulitan itu muncul dan untuk mengetahui bantuan apa yang dibutuhkan oleh yang bersangkutan. Berdasarkan data hasil asesmen tersebut dapat dibuat program pembelajaran yang tepat bagi anak itu.

2.2.3 *Progressive Web App*

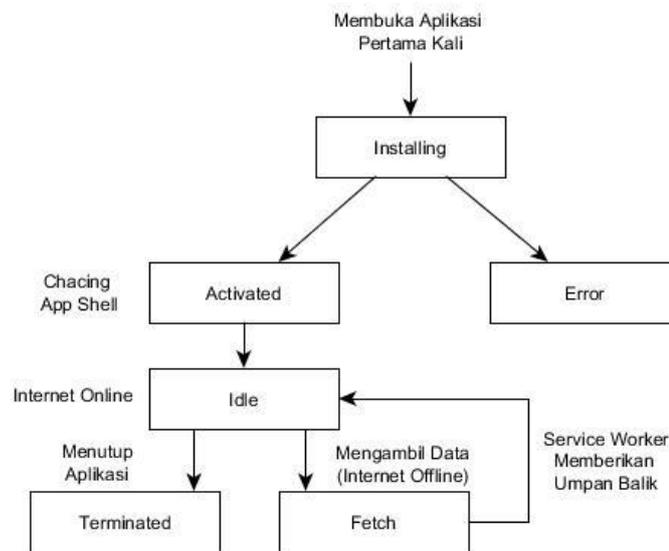
Osmani menyebutkan “*PWAs are defined by a set of concepts and keywords including progressive, responsive, connectivity independent, app-like, fresh, safe, discoverable, reengageable, installable, and linkable*” [13]. Artinya, *Progressive Web Apps* (PWA) didefinisikan sebagai serangkaian konsep dan kata kunci seperti progresif, responsif, konektivitas yang independen, menyerupai aplikasi *native*, segar, aman, mudah ditemukan, dapat direkayasa ulang, dapat dipasang, dan ditautkan.

Progressive Web Apps (PWA) adalah konsep pengalaman pengguna yang menggabungkan bagian terbaik web dan bagian terbaik *native* app. PWA berguna untuk pengguna sejak pertama membuka halaman sebuah web dengan konsep PWA, dan seiring dengan pengguna menggunakan aplikasi web lebih banyak lagi, aplikasi akan menjadi semakin *powerful*. Aplikasi dapat dimuat dengan cepat, bahkan dalam kondisi internet yang kurang baik sekalipun, dapat mengirim *push notifications*, mempunyai ikon aplikasi di *home screen*, dan bisa berjalan dalam mode layar penuh [5].

Adapun beberapa teknologi yang ada dalam *Progressive Web App* adalah sebagai berikut :

a. *Service Worker*

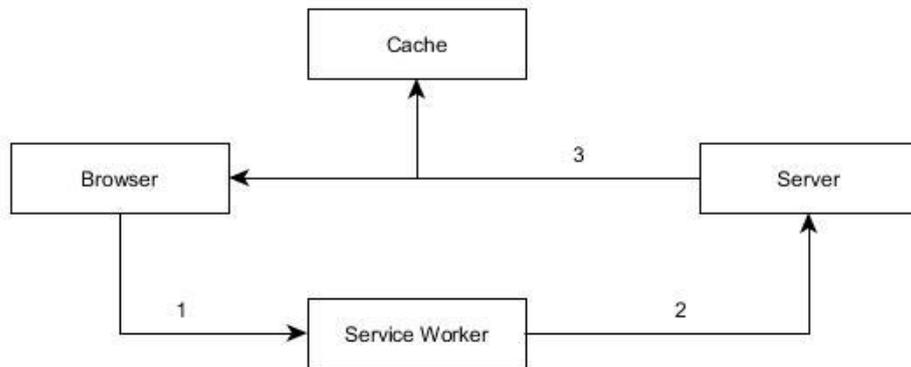
Service Worker adalah salah satu jenis dari *web worker*, yaitu script yang berjalan di belakang browser pengguna. *Service Worker* pada dasarnya adalah berkas JavaScript yang berjalan pada *thread* yang berbeda dengan *main thread* browser, menangani *network request*, *caching*, mengembalikan *resource* dari *cache*, dan bisa mengirimkan *push message* [14]. Berikut daur hidup (*life cycle*) *Service Worker*:



Sumber gambar : Platform E-Learning untuk Pembelajaran
 Pemrograman Web Menggunakan Konsep Progressive Web Apps [14]

Gambar 2.3 Life cycle Service woker

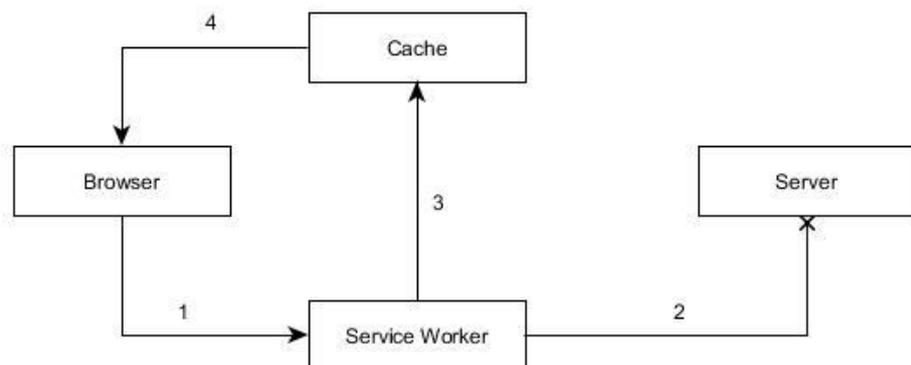
Service Worker bekerja sebagai jembatan setiap *request* yang akan dikirimkan kepada web server. *Service Worker* akan memeriksa setiap *request* yang dikirimkan apakah tersedia di *cache* atau tidak, jika tersedia maka *request* yang dikirimkan akan diteruskan kepada *cache*, sebaliknya jika tidak maka *request* akan diteruskan kepada web server. Berikut adalah gambaran alur kerja *Service Worker* dalam kondisi jaringan *online* :



Sumber gambar : Platform E-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps [14]

Gambar 2.4 Service Worker dalam jaringan Online

Disamping itu *Service Worker* akan menduplikasi setiap *request* baru yang dibuat dan menyimpannya ke dalam *cache* browser yang setelahnya akan diteruskan kepada halaman web, hal ini dapat membuat setiap halaman yang sebelumnya dikunjungi dapat tersedia dalam jaringan *offline*. Berikut adalah gambaran alur kerja *Service Worker* dalam kondisi jaringan *offline* :



Sumber gambar : Platform E-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps [14]

Gambar 2.5 Service Worker dalam Kondisi Offline

b. *Web App Manifest*

Web App Manifest adalah salah satu dari kumpulan teknologi dalam *Progressive Web App*. *Web App Manifest* berbentuk file JSON sederhana yang berfungsi sebagai penyedia informasi aplikasi web. Informasi yang disediakan seperti nama aplikasi, pemilik, ikon, dan deskripsi aplikasi web, hal ini bertujuan agar aplikasi web yang dibuat dapat memberikan fitur *Add To Homescreen*.

c. *IndexedDB*

IndexedDB adalah media penyimpanan lokal *non-sql* yang ada pada browser. *IndexedDB* memungkinkan kita menyimpan komponen-komponen yang ada pada aplikasi pada saat jaringan internet dalam kondisi *offline*, yang mana hal ini sangat bagus untuk memastikan setiap *request* yang dikirim dapat tersimpan didalam *database* aplikasi.

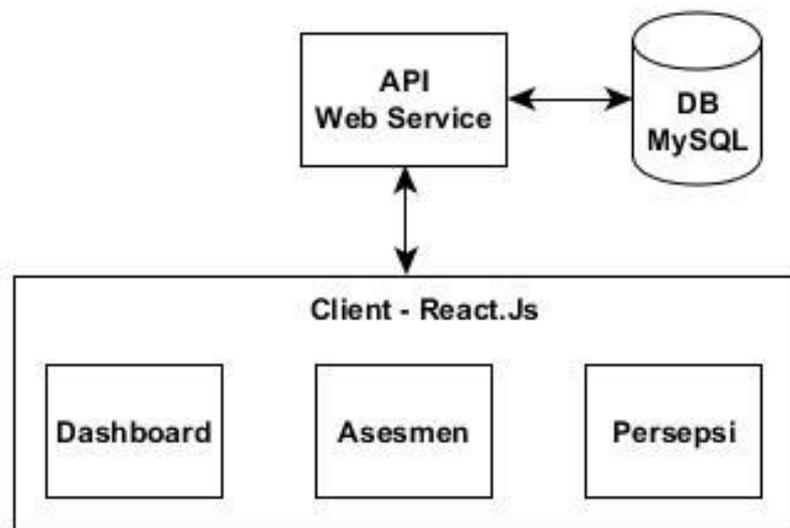
d. *BackgroundSync*

BackgroundSync adalah API baru yang memungkinkan aplikasi menunda tindakan hingga pengguna memiliki jaringan internet yang stabil. Hal ini berguna untuk memastikan apapun yang dikirim oleh pengguna telah berhasil terkirim.

2.2.4 *React.js*

React.js adalah *UI library* yang dikembangkan oleh Facebook untuk memfasilitasi pengembang web interaktif, *stateful*, dan komponen *UI* yang mudah digunakan. *React.js* sudah digunakan pada *website* Facebook dibagian produksi. *React.js* juga merupakan *library* terbaik untuk melakukan *render user interface* yang kompleks dengan performa yang tinggi. Konsep dasar pada *React.js* adalah ada pada *Virtual DOM*. *React.js* lebih efektif menggunakan *Virtual DOM*, yang mana dapat melakukan *render* pada bagian *client* ataupun bagian server [15].

React.js berinteraksi dengan menggunakan API sebagai jembatan agar *React.js* dapat berinteraksi dengan *database*. Berikut adalah alur atau struktur *React.js* berkomunikasi dengan *database*.



Gambar 2.6 Struktur komunikasi *React.js*

2.2.5 API

Saputra [16] mengatakan bahwa API adalah aplikasi pemrograman yang secara khusus dikembangkan untuk digunakan sebagai perantara komunikasi antara komponen-komponen perangkat lunak. Biasanya, hasil output dari API dapat berupa data *XML* ataupun *JSON*, tergantung dari situs mana yang menyediakan API tersebut [16].

Dengan adanya API, maka memudahkan seorang *programmer* untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan *developer* menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi. Keuntungan dengan menggunakan API adalah sebagai berikut [17]:

1. Portabilitas

Developer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah terinstal API tersebut.

2. Lebih Mudah Dimengerti

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan.

2.2.6 *CodeIgniter*

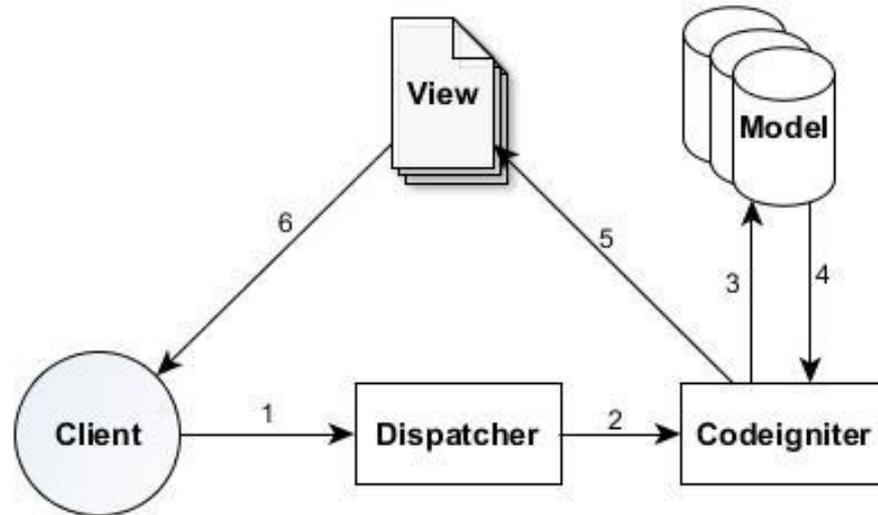
CodeIgniter adalah sebuah *framework* untuk web yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi web yang kompleks. *CodeIgniter* dapat mempercepat proses pembuatan web, karena semua *class* dan modul yang dibutuhkan sudah ada dan *programmer* hanya tinggal menggunakannya kembali pada aplikasi web yang akan dibuat [18].

Tujuan utama dari *CodeIgniter* adalah untuk memudahkan *programmer* dalam mengembangkan aplikasi secara cepat tanpa harus melakukan pemrograman dari nol. Dengan demikian, *programmer* PHP yang telah memiliki dasar pemrograman, dapat bekerja lebih cepat menghasilkan aplikasi yang dibutuhkan.

Ada beberapa alasan mengapa menggunakan *Framework*:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada). Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple *database*, scaffolding, pengaturan session, *Error handling*, dll)
3. Lebih bebas dalam pengembangan jika dibandingkan CMS.

Konsep MVC *Framework* dapat dilihat pada gambar dibawah ini :

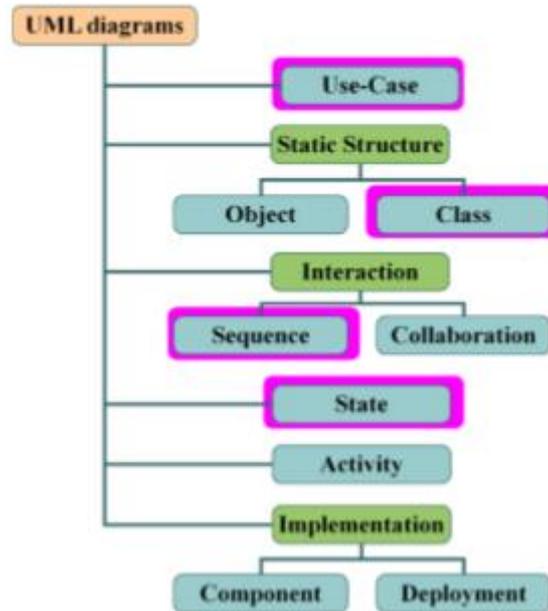


Gambar 2.7 Konsep MVC Framework CodeIgniter

2.2.7 UML

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET [19].

Berikut gambar dari diagram UML



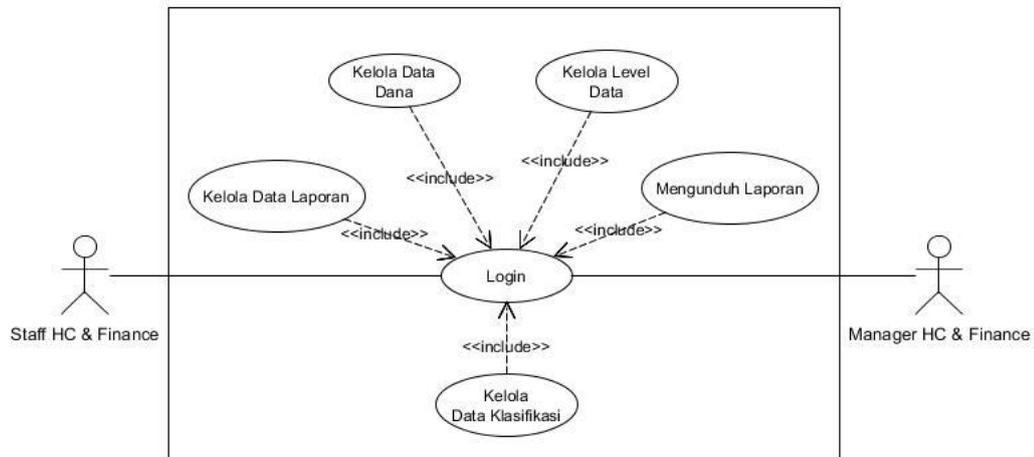
Sumber gambar : Memahami Penggunaan UML (Unified Modelling Language) [20].

Gambar 2.8 Struktur diagram UML

Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut yang berbeda - beda sehingga bisa mendapatkan pemahaman secara menyeluruh. Untuk upaya tersebut UML menyediakan beberapa jenis diagram yang dapat dikelompokkan berdasarkan sifatnya statis atau dinamis. Berikut ini adalah beberapa bagian dalam diagram UML :

a. Use case Diagram

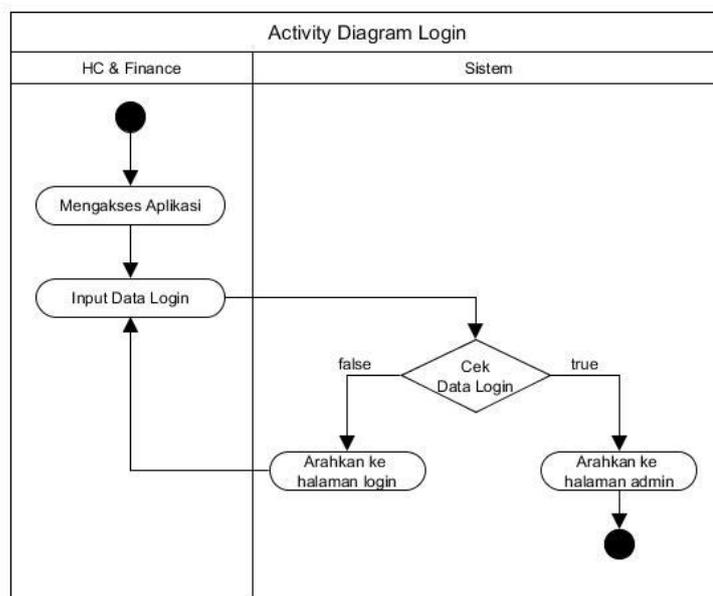
Diagram ini memperlihatkan himpunan *Use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.



Gambar 2.9 Contoh Use case Diagram

b. Activity Diagram

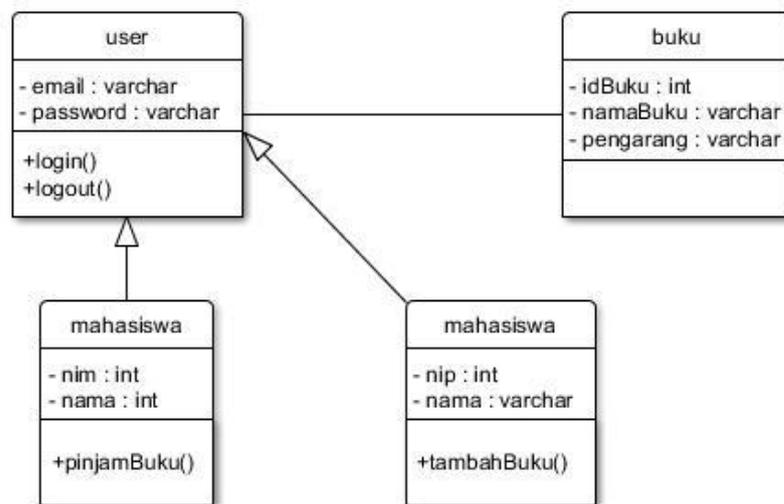
Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.



Gambar 2.10 Contoh Activity Diagram

c. Class Diagram

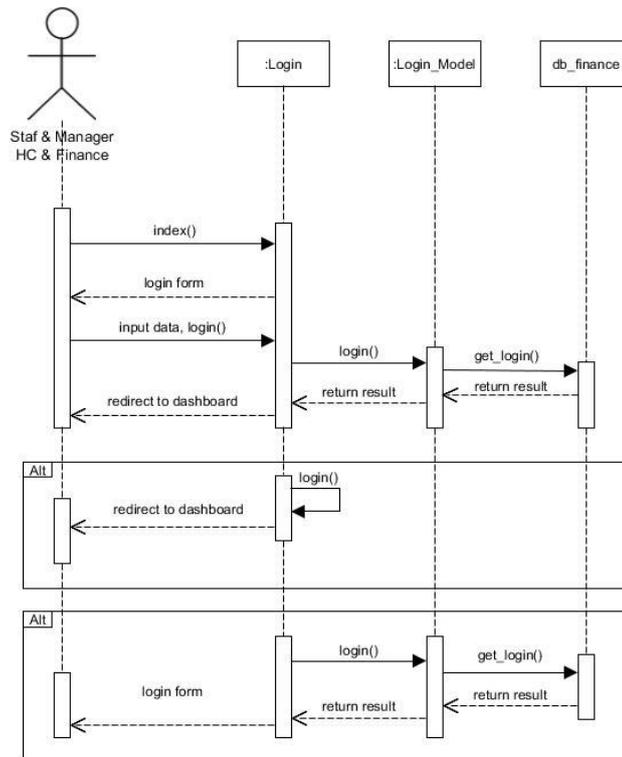
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class* diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class* diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.



Gambar 2.11 Contoh Class Diagram

d. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *Use case* diagram.



Gambar 2.12 Contoh Sequence Diagram

2.2.8 Lighthouse

Pada awal tahun 2018, Google merilis aplikasi *SEO* yang bernama *Lighthouse*. *Lighthouse* yang menawarkan bagaimana cara meningkatkan kualitas halaman web. *Lighthouse* adalah aplikasi *open source* yang dibangun oleh Google untuk menguji [5]:

- *Progressive Web App*,
- *Performance*,
- *Accessibility*,
- *Best Practices*,
- *SEO*.

Hasil pengujian menggunakan *Lighthouse*, aplikasi web yang dibangun bisa dikatakan sudah memenuhi kriteria sebuah *Progressive Web Apps* dengan nilai 100 dari 100.

Aplikasi juga sudah mendapat nilai yang baik pada ketiga kriteria lainnya, kriteria *performance* dengan rata-rata 85 dari 100, kriteria *accessibility* dengan rata-rata 97 dari 100, dan kriteria *best practices* dengan rata-rata 100 dari 100.

Setelah proses audit selesai pengguna dapat mengambil informasi ini dan memeriksa setiap indikator yang gagal. Berikut adalah penjelasan detail mengenai indikator-indikator yang ditampilkan oleh *Google Lighthouse*:

a. *Progressive Web App*

Progressive Web App dinilai berdasarkan 3 faktor yaitu kecepatan, keandalan, dan keterlibatan *PWA* pada aplikasi tersebut. Audit *Google Lighthouse* juga menyediakan daftar ceklis indikator *PWA* sehingga pengguna tahu bagian mana indikator *PWA* yang belum optimal.

b. *Performance*

Lighthouse memberikan laporan kinerja situs dalam beberapa kategori yaitu metrik, peluang, dan diagnostik. Metrik mencakup tiga poin utama dari proses pemuatan, menunjukkan kepada pengguna fitur halaman mana yang memperlambat proses. Selanjutnya pengguna akan diberikan beberapa rekomendasi untuk mempercepat kinerja, beberapa rekomendasi ini mencakup seperti kompresi gambar, dan mengoptimalkan sumber daya lainnya.

c. *Accessibility*

Fitur audit *Google Lighthouse* ini memeriksa perilaku pengguna non-tradisional. *Accessibility* ditujukan untuk pengunjung yang menderita disabilitas. Bagian alat audit ini menganalisis pembaca layar dan teknologi bantu lainnya di halaman.

d. Best Practices

Google memiliki banyak aturan terhadap setiap web yang akan di indeks. Pengujian ini berfungsi untuk memastikan bahwa aplikasi web telah mengikuti aturan dan penerapan praktik yang terbaik. Fitur ini juga membantu pengguna menghindari kinerja yang gagal dan memodernisasi halaman.

e. SEO

Google Lighthouse juga melakukan pemeriksaan pada bagian *Search Engine Optimization*. Hal ini sangat penting apabila aplikasi web diharapkan dapat mendatangkan banyak pengunjung. Dengan indikator ini pengguna dapat mengoptimalkan peringkat aplikasi web dalam mesin pencari Google.

