

1. Sejarah Aplikasi

Aplikasi adalah sebuah perangkat lunak atau program yang diciptakan dan dikembangkan untuk melakukan tugas-tugas tertentu pada perangkat komputer, laptop ataupun smartphone. Aplikasi berasal dari bahasa Inggris Application yang artinya penerapan atau penggunaan. Secara teknis back-end aplikasi dibuat oleh para *programmer* atau *developer* dengan menggunakan bahasa pemrograman tertentu.

Setelah selesai, front-end aplikasi dapat digunakan oleh para user dengan menginstalnya pada perangkat elektronik agar dapat digunakan untuk berbagai keperluan, seperti mengolah dokumen, komunikasi, desain grafis, manajemen perangkat keras, bermain game dan lain sebagainya.

Aplikasi sendiri terbagi menjadi 3 jenis, yakni, Aplikasi dekstop (1) yang dijalankan pada komputer atau pc. Aplikasi web (2) yang dijalankan menggunakan komputer dengan koneksi internet dan Aplikasi mobile (3) yang dapat dijalankan di perangkat mobile.

2. Bahasa Pemrograman

Bahasa pemrograman adalah bahasa buatan yang digunakan untuk mengendalikan perilaku dari sebuah mesin, biasanya berupa mesin komputer, sehingga dapat digunakan untuk memberitahu komputer tentang apa yang harus dilakukan. Komputer “berpikir” dalam biner – string 1 dan 0. Bahasa pemrograman memungkinkan kita menerjemahkan angka 1 dan 0 menjadi sesuatu yang dapat dipahami dan ditulis oleh manusia. Bahasa pemrograman sebagai jembatan yang memungkinkan manusia menerjemahkan pikiran kita ke dalam instruksi yang dapat dipahami komputer. Dengan adanya bahasa pemrograman kita dapat membuat situs website, aplikasi, permainan, dan sistem dengan menggunakan bahasa pemrograman.

2.1. Fungsi Bahasa Pemrograman

2.1.1. Media Komunikasi antara Manusia dan Komputer

Fungsi pertama bahasa pemrograman adalah sebagai jembatan komunikasi antara programmer dan komputer. Ini sesuai dengan fungsi bahasa itu sendiri, yakni alat komunikasi yang dipahami oleh penuturnya. Bahasa yang dipahami komputer berbeda dengan bahasa manusia. Maka dari itu, perintah bahasa pemrograman yang ditulis programmer harus diterjemahkan terlebih dahulu, agar bisa dimengerti komputer. Untuk menerjemahkan bahasa pemrograman ke bahasa komputer, ada dua metode yang biasa digunakan, yaitu:

a. Metode Compiler

Compiler adalah tools untuk memindai seluruh perintah bahasa pemrograman, lalu mengubahnya menjadi sebuah paket file baru. Paket file ini berisikan kode mesin yang dapat dibaca komputer atau perangkat lain.



Kemudian, komputer dapat mengeksekusi perintah bahasa pemrograman setiap kali pengguna menjalankan paket file tersebut. Ekstensi paket file ini bisa berbeda, sesuai dengan perangkat yang akan menjalankan program. Misalnya, .EXE untuk perangkat berbasis Windows, .APK untuk Android, atau .APP untuk iOS. Sementara, bahasa pemrograman yang tercatat menggunakan Compiler antara lain C, C++, C#, Java, Kotlin, dan Swift.

b. Metode Interpreter

Interpreter adalah tools untuk mengubah perintah bahasa pemrograman menjadi kode mesin, tanpa menghasilkan paket file baru. Caranya dengan menerjemahkan perintah tersebut sebaris demi sebaris, saat program dijalankan.



Browser adalah interpreter yang umum digunakan. Oleh sebab itu, Interpreter sering digunakan oleh bahasa pemrograman spesialis website, seperti JavaScript, Ruby, dan PHP.

2.1.2. Alat untuk Mengembangkan Aplikasi

Fungsi kedua bahasa pemrograman adalah sebagai tools untuk mengembangkan sistem dan program. Cara kerjanya, programmer akan menulis kode program (coding) yang kemudian diubah menjadi kode mesin.

2.2. Jenis-jenis Bahasa Pemrograman

2.2.1. Bahasa Pemrograman Tingkat Rendah

Bahasa tingkat rendah adalah jenis bahasa pemrograman yang paling dekat hubungannya dengan logika komputer. Bahasa jenis ini bisa dijalankan di komputer tanpa perlu diterjemahkan terlebih dahulu dengan Compiler atau Interpreter. Bahasa pemrograman tingkat rendah ini dibagi menjadi dua:

a. Bahasa Mesin

Bahasa mesin adalah bahasa asli yang dipahami komputer. Bahasa ini juga dikenal sebagai kode mesin atau kode objek. Umumnya, bahasa mesin ditampilkan dalam kode biner (0 dan 1).

Example of machine-language

Here's what a program-fragment looks like:

```
10100001 10111100 10010011 00000100
00001000 00000011 00000101 11000000
10010011 00000100 00001000 10100011
11000000 10010100 00000100 00001000
```

It means: $z = x + y;$

Oleh karena itu, bahasa mesin cenderung sulit dipahami oleh manusia. Namun program yang ditulis dengan bahasa mesin punya kinerja yang sangat gesit, karena tidak perlu diterjemahkan lagi. Bahasa mesin merupakan hasil akhir dari proses penerjemahan bahasa tingkat tinggi yang dilakukan oleh compiler dan interpreter.

b. Bahasa Assembly

Bahasa Assembly adalah jenis bahasa tingkat rendah untuk membangun sistem yang berhubungan langsung dengan perangkat keras. Contohnya untuk membuat sistem operasi atau driver.

```
section      .text
global      _start                ;must be declared for linker (ld)

_start:     ;tell linker entry point

    mov     edx,len                ;message length
    mov     ecx,msg                ;message to write
    mov     ebx,1                  ;file descriptor (stdout)
    mov     eax,4                  ;system call number (sys_write)
    int     0x80                  ;call kernel

    mov     eax,1                  ;system call number (sys_exit)
    int     0x80                  ;call kernel

section      .data
msg          db 'Hello, world!',0xa ;our dear string
len          equ $ - msg           ;length of our dear string
```

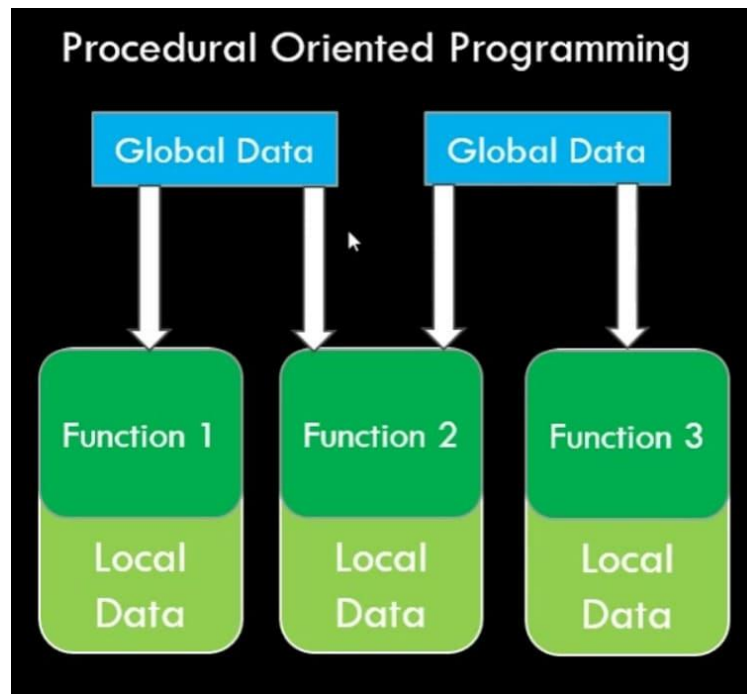
Proses penerjemahan ini dilakukan sebuah tools bernama Assembler. Meskipun harus diterjemahkan terlebih dahulu, program Assembly memiliki performa yang cepat karena tidak membutuhkan banyak memory saat menjalankannya.

2.2.2. Bahasa Pemrograman Tingkat Tinggi

Bahasa tingkat tinggi adalah jenis bahasa pemrograman dengan perintah yang menyerupai bahasa manusia. Dengan kata lain, perintah tersebut sangat mudah ditulis, dibaca, dan dikelola untuk keperluan lebih lanjut. Bahasa tingkat tinggi sering dimanfaatkan untuk membangun website dan aplikasi. Bahkan, hampir semua website dan aplikasi yang Anda install dibuat menggunakan bahasa tingkat tinggi. Bahasa tinggi perlu diterjemahkan menjadi Bahasa mesin agar dapat berjalan melalui proses compiler dan interpreter. Bahasa pemrograman tingkat tinggi dibagi menjadi dua:

a. Pemrograman Berorientasi Prosedural (Procedural Oriented Programming)

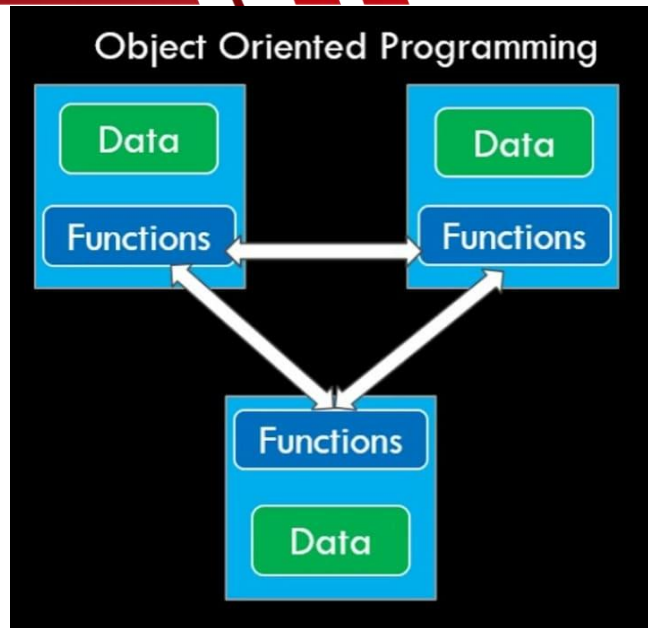
Pemrograman berorientasi prosedural adalah jenis bahasa pemrograman yang membagi perintah program menjadi bagian-bagian yang lebih kecil. Bagian tersebut akan diurutkan secara sistematis. Komputer nantinya akan melaksanakan perintah sesuai dengan urutan yang sudah ditetapkan.



bahasa pemrograman berorientasi prosedural adalah programmer akan lebih mudah memahami alur program. Namun kekurangannya, akan lebih sulit memodifikasi satu bagian program tanpa mempengaruhi bagian yang lainnya. Contoh bahasa pemrograman berorientasi prosedural adalah Pascal, Basic, dan FORTRAN.

b. Pemrograman Berorientasi Objek (Object Oriented Programming)

Pemrograman berorientasi objek adalah jenis bahasa pemrograman yang mengelola perintah program menjadi objek-objek tertentu. Masing-masing objek akan memiliki karakteristiknya tersendiri. Komputer akan mengeksekusi perintah yang diberikan sesuai dengan karakteristik masing-masing objek.



Keuntungan dari pemrograman berorientasi objek adalah kemudahan dari segi modifikasi, tanpa menyebabkan error di bagian program lain. Kekurangannya, program jadi lebih sulit dipahami alurnya. Contoh bahasa pemrograman berorientasi objek antara lain Java, PHP, Python, dan Ruby.

2.3. Contoh-contoh Bahasa Pemrograman

Terdapat banyak Bahasa pemrograman tingkat tinggi yang dapat digunakan untuk membuat program dan sistem. Pemilihan Bahasa pemrograman tergantung dengan program atau sistem yang akan dibuat baik berupa web, desktop, ataupun mobile. Berikut beberapa Bahasa pemrograman yang populer digunakan untuk membuat program:

a. Javascript



JavaScript adalah bahasa pemrograman yang dipakai untuk mengembangkan website dari segi klien (client side). JavaScript merupakan salah satu bahasa tertua di dunia, karena sudah dikembangkan sejak tahun 1995. JavaScript memungkinkan programmer menambahkan fitur-fitur kompleks sehingga halaman website jadi lebih dinamis dan interaktif. Contohnya untuk membuat pop up notifikasi, pesan konfirmasi, bahkan games sederhana di halaman web.

Bahkan, kemampuan JavaScript semakin meningkat berkat adanya Node.js, sebuah runtime environment. Jadinya, JavaScript kini tidak hanya dipakai untuk membangun website dinamis. Ia juga dimanfaatkan dalam pengembangan server, aplikasi mobile, animasi grafis, games, dan masih banyak lagi.

Kelebihan bahasa pemrograman JavaScript:

- Membuat website jadi lebih dinamis, sekaligus menambah fitur-fitur baru di website.
- Mempersingkat waktu loading halaman website dari server.
- Memungkinkan untuk digunakan bersama bahasa lain.
- Merupakan bahasa pemrograman yang sangat mudah dipahami dan dipelajari pemula.
- Mampu mengembangkan sistem yang solid karena ekosistemnya yang luas dan fleksibel.

Kekurangan bahasa pemrograman JavaScript:

- JavaScript murni tidak bisa berjalan di luar browser.
- Perintah JavaScript cenderung kurang aman dibanding bahasa pemrograman lain.
- Ekosistem yang terlalu luas berpotensi membingungkan programmer pemula.

b. Python



Python adalah bahasa pemrograman open source populer yang cukup mudah dipahami pemula. Sama seperti JavaScript, Python sudah ada sejak dekade 90-an, tepatnya di tahun 1991. Bahasa Python sering disebut sebagai bahasa multiguna, karena bisa dipakai untuk berbagai jenis sistem. Tapi kebanyakan, Python dipakai untuk membangun Machine Learning, Artificial Intelligence, Big Data, dan Robotika.

Kelebihan bahasa pemrograman Python:

- Perintah Python sangat mudah dipahami, dibaca, dan ditulis.
- Sanggup terintegrasi dengan bahasa pemrograman lain, seperti C, C++, dan Java.

- Python mengeksekusi perintahnya baris demi baris, sehingga programmer bisa menemukan error lebih mudah.
- Perintah Python bisa dijalankan di berbagai platform tanpa terkendala masalah kompatibilitas, istilahnya Write Once Run Anywhere.

Kekurangan bahasa pemrograman Python:

- Perintah Python diterjemahkan oleh Interpreter, sehingga performanya tidak sebaik bahasa yang menggunakan Compiler.
- Interpreter juga membuat Python kurang cocok untuk membangun aplikasi mobile atau games.

c. PHP



Bahasa pemrograman PHP adalah salah satu bahasa pemrograman paling populer saat ini. Bahasa PHP juga tergolong veteran, karena sudah dikembangkan sejak tahun 1994, oleh Rasmus Laird. Bahasa yang satu ini sangat kompatibel, dan lazim digunakan bersama bahasa lain seperti JavaScript, HTML, CSS, bahkan SQL. PHP paling sering dimanfaatkan untuk membangun website dan aplikasi berbasis web.

Kelebihan bahasa pemrograman PHP:

- Bahasa PHP sangat mudah dipelajari pemula, dan perintahnya tergolong aman.
- Kemampuannya bisa ditingkatkan dengan memasang library, modul, dan framework PHP tambahan.
- Dapat berjalan di berbagai sistem operasi, sebut saja Windows, Mac, Linux, atau Android.
- Kompatibel dengan berbagai layanan Cloud, seperti Amazon Web Services.

Kekurangan bahasa pemrograman PHP:

- Belum teruji kemampuannya untuk membangun sistem selain yang berbasis web.
- Performanya tidak begitu stabil ketika dipaksa mengelola aplikasi berskala besar.

- Harus dipasangkan dengan bahasa pemrograman lain agar hasil akhir sistem lebih optimal.

d. Java



Java adalah bahasa pemrograman multiplatform dilengkapi performa stabil serta keamanan terjamin. Sama seperti JavaScript dan Python, Java juga tergolong veteran karena pertama kali diluncurkan tahun 1995. Mengingat keamanannya, Java banyak dipakai untuk membangun sistem informasi perbankan, retail, dan pemerintahan. Selain itu, Java banyak dimanfaatkan dalam pengembangan aplikasi mobile dan desktop.

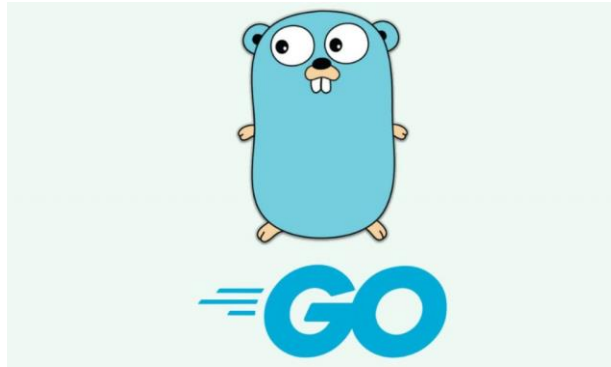
Kelebihan bahasa pemrograman Java:

- Perintah Java cukup mudah dipahami, ditulis, dan diperbaiki ketika terjadi error.
- Program Java dapat dikonversi dan berjalan di berbagai platform.
- Hasil akhir perintah Java adalah kode mesin murni, sehingga lebih aman dari kemungkinan diintip penyusup.
- Mampu menjalankan beberapa perintah berbeda di waktu yang bersamaan.

Kekurangan bahasa pemrograman Java:

- Perintah Java cenderung berbelit-belit, alias butuh beberapa baris kode untuk menulis satu perintah sederhana.
- Banyaknya baris kode ini membuat program Java memakan banyak memori ketika dijalankan.
- Java tidak menyediakan fasilitas backup secara bawaan.

e. Go



Go atau Golang adalah bahasa pemrograman open source yang diterjemahkan oleh Compiler. Bahasa ini mulai dikenalkan pada tahun 2009 oleh raksasa teknologi dunia, Google. Bahasa ini menggabungkan beberapa keunggulan dari bahasa lain, seperti kecepatan performa bahasa C dan kemudahan dari JavaScript dan Python. Bahasa Go banyak digunakan untuk membangun server dan aplikasi berskala besar.

Kelebihan bahasa pemrograman Go:

- Bahasa Go lumayan mudah dipahami dan dipelajari pemula.
- Sudah menyediakan tools bawaan untuk proses testing aplikasi.
- Performa yang terbilang cepat, karena menggunakan Compiler.

Kekurangan bahasa pemrograman Go:

- Belum mendukung metode penanganan error yang mumpuni.
- Masih belum populer, jumlah komunitasnya juga belum banyak.
- Ekosistem yang masih perlu waktu untuk berkembang.

f. Kotlin



Kotlin adalah bahasa multiplatform dan open source yang dikembangkan oleh perusahaan JetBrains pada tahun 2011 lalu. Meski multiplatform, Kotlin paling cocok dipakai untuk membangun aplikasi Android. Alasannya, Kotlin didukung secara resmi oleh Google sebagai bahasa pengantar untuk pengembangan Android.

Kelebihan bahasa pemrograman Kotlin:

- Perintah bahasa Kotlin cenderung ringkas, sehingga bisa menghemat baris kode.

- Dukungan resmi Google untuk pengembangan fitur, peningkatan kinerja, dan pengamanan sistem secara berkelanjutan.
- Perintah program yang lebih aman dari kemungkinan error, karena aturan coding yang mewajibkan deklarasi variabel di awal.
- Bisa digunakan bersama Java untuk mengembangkan satu aplikasi yang sama.

Kekurangan bahasa pemrograman Kotlin:

- Kecepatan proses Compile yang cenderung tidak stabil.
- Jumlah komunitas programmer Kotlin masih minim.
- Kemampuan Kotlin dalam membangun aplikasi di platform selain Android masih belum teruji.

g. Swift



Jika Kotlin adalah bahasa pengantar untuk Android, maka Swift adalah bahasa pengantar untuk iOS. Bahasa ini dikembangkan secara eksklusif oleh Apple sejak tahun 2014 lalu. Mirip dengan Dart, Swift harus ditandemkan dengan sebuah UI toolkit bernama Xcode. Bedanya, Xcode ini berbayar dan hanya dapat diinstall di perangkat buatan Apple. Maka bisa ditebak, bahasa pemrograman Swift khusus dipakai di perangkat Apple, untuk mengembangkan aplikasi berbasis Apple juga, seperti iOS dan Mac OS.

Kelebihan bahasa pemrograman Swift:

- Dukungan resmi dari Apple untuk pengembangan ekosistem.
- Cukup mudah dipahami untuk programmer pemula.
- Performanya diklaim sangat cepat dibanding bahasa pemrograman lain.

Kekurangan bahasa pemrograman Swift:

- Harus dipasangkan dengan Xcode, tidak bisa berdiri sendiri.
- Swift memang open source, tapi Xcode itu sendiri berbayar, sehingga menyulitkan sebagian programmer.
- Harus mempunyai perangkat Apple untuk belajar pemrograman Swift.

h. Dart



Dart adalah bahasa resmi untuk menggunakan UI toolkit gratisan bernama Flutter. Kombinasi keduanya dapat dimanfaatkan untuk mengembangkan aplikasi berbasis Android, iOS, dan web dengan perintah program yang sama.

Kelebihan bahasa pemrograman Dart:

- Perintah program menyerupai JavaScript, sehingga mudah dipelajari bagi yang sudah menguasai JavaScript.
- Performa yang cenderung lebih gesit, dibandingkan dengan JavaScript.
- Dokumentasi yang terbilang lengkap, sehingga tidak menyulitkan programmer pemula.

Kekurangan bahasa pemrograman Dart:

- Harus dipasangkan dengan Flutter, alias tidak bisa digunakan dengan tools lain.
- Masih belum begitu populer, berbanding lurus dengan komunitas yang minim.
- Masih dalam proses perkembangan, sehingga dokumentasi yang ada saat ini bisa saja berubah lagi.

i. SQL



SQL adalah bahasa pemrograman yang berguna untuk mengakses, mengelola, dan memanipulasi data berbasis relasional. SQL ini pertama kali dikembangkan oleh IBM pada tahun 1974. Sebagai bahasa pemrograman khusus data, ia sangat berguna dalam mendefinisikan struktur data dan alirannya dalam suatu program. Tanpa SQL, data program tidak dapat diolah dengan maksimal. Maka dari itu, SQL banyak dimanfaatkan untuk membangun sistem manajemen database (DBMS) untuk aplikasi dengan platform apapun.

Kelebihan bahasa pemrograman SQL:

- Performa pengolahan data yang sangat cepat, tidak sampai hitungan detik.
- Perintah SQL paling mirip dengan bahasa manusia, sehingga mudah dipelajari.
- Ekosistem yang terjamin, karena sudah ada sejak dekade 70-an.
- Kompatibel dengan berbagai platform dan bahasa pemrograman lain.

Kekurangan bahasa pemrograman SQL:

- Meski perintahnya mudah dipahami, interface SQL sendiri terbilang kompleks sehingga kurang ramah pengguna.
- Kemampuan yang terbatas, hanya berguna untuk mengelola data.
- Tidak bisa berdiri sendiri, harus digunakan bersama bahasa pemrograman lain jika ingin membangun sebuah sistem utuh.

3. Back End dan Front End

Pada pengembangan aplikasi berbasis web tidak jarang kita jumpai istilah backend dan frontend. Keduanya saling berkaitan dan bertanggung jawab bagaimana aplikasi web dapat berfungsi dan diakses oleh pengguna.

3.1. Frontend

Merupakan bagian dari aplikasi web yang berinteraksi langsung dengan pengguna. Biasanya juga disebut dengan client-side sebuah aplikasi. Hal ini mencakup semua yang dialami pengguna secara langsung: warna dan gaya teks, gambar, grafik dan tabel, tombol, warna, dan menu navigasi. HTML, CSS, dan JavaScript adalah bahasa yang digunakan untuk pengembangan frontend. Struktur, desain, perilaku, dan konten dari semua yang terlihat di layar browser saat situs web, aplikasi web, atau aplikasi seluler dibuka, diimplementasikan oleh frontend developer. Responsif dan kinerja adalah dua tujuan utama frontend. Developer harus memastikan bahwa situs tersebut responsif, yaitu muncul dengan benar pada perangkat dari semua ukuran, tidak ada bagian dari situs web yang berperilaku tidak normal terlepas dari ukuran layar. Ada beberapa framework/ library frontend yang populer digunakan diantaranya React, Vue, jQuery, SASS, Flutter, Tailwind, Bootstrap dll.

3.2. Backend

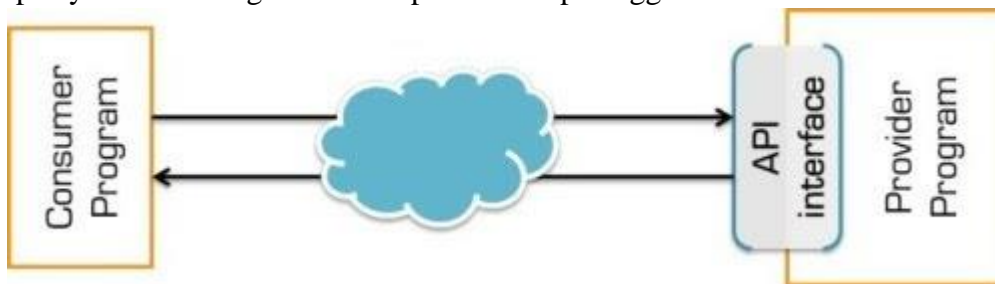
Backend adalah server-side dari situs web. Backend menyimpan dan mengatur data, dan juga memastikan semua yang ada di client-side situs web berfungsi dengan baik. Backend adalah bagian dari situs web yang tidak terlihat dan tidak bersentuhan langsung dengan pengguna. Bagian dan karakteristik yang dikembangkan oleh desainer backend diakses secara tidak langsung oleh pengguna melalui aplikasi frontend. Backend sebuah aplikasi bisa dibangun menggunakan bahasa pemrograman seperti PHP, Node.js, Java, Go, Ruby, dll. Untuk mempercepat proses pengembangan, pengembang bisanya juga menggunakan framework/ library sesuai dengan bahasa

pemrograman yang digunakan. Untuk bahasa pemrograman Node.js framework yang populer digunakan adalah express, hapi, adonis, nest, dll. Sedang untuk PHP terdapat laravel, codeigniter, yii, dll.

4. API

4.1. Pengertian

API adalah singkatan dari Application Programming Interface. API merupakan interface yang dapat menghubungkan dua aplikasi agar dapat saling berkomunikasi atau bertukar data. Kita dapat menganalogikan sebuah API sebagai seorang pelayan restoran dan pengguna sebagai pelanggan. Seorang pelayan akan menyampaikan pesanan pelanggan ke dapur atau juru masak. Jika hidangan sudah selesai maka pelayan akan menghantarkan pesanan ke pelanggan.



API dapat dibangun menggunakan bahasa pemrograman berbasis backend. API dapat digunakan untuk berkomunikasi dengan berbagai bahasa pemrograman yang berbeda. Hal tersebut tentunya akan mempermudah developer dalam mengembangkan aplikasi dalam berbagai bentuk mulai dari web, mobile maupun desktop.

4.2. Arsitektur API

Secara umum terdapat tiga arsitektur yang dapat digunakan developer untuk membangun API. Arsitektur API yang dimaksud adalah bentuk data yang digunakan untuk berkomunikasi. Arsitektur tersebut adalah:

a. RPC

Remote Procedure Calls (RPC) RPC adalah suatu protokol yang menyediakan suatu mekanisme komunikasi antar proses yang memungkinkan suatu program untuk berjalan pada suatu komputer tanpa terasa adanya eksekusi kode pada sistem yang jauh (remote system). RPC memiliki dua jenis XML-RPC dan JSON-RPC.

b. SOAP

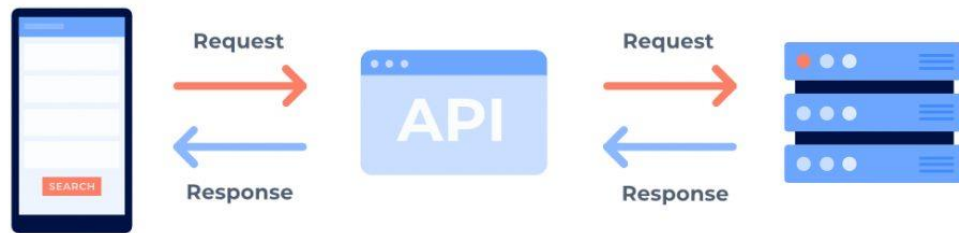
Arsitektur API lainnya adalah SOAP (Simple Object Access Protocol). Arsitektur ini menggunakan XML (Extensible Markup Language) yang memungkinkan semua data disimpan dalam dokumen.

c. REST

REST adalah singkatan dari REpresentational State Transfer. REST merupakan gaya arsitektur untuk menyediakan standar antara sistem komputer di web, sehingga memudahkan sistem untuk berkomunikasi satu sama lain. Format data yang digunakan REST adalah JSON.

4.3. Cara Kerja API

Untuk mengetahui cara kerja sebuah API bisa dilihat gambar berikut:



1. Aplikasi mengakses API

Bagian pertama adalah ketika pengguna mengakses alamat API atau biasa disebut endpoint sesuai dengan fungsi/ fitur. Saat mengakses endpoint API, aplikasi juga akan mengirimkan parameter/ data.

2. API melakukan request ke server

Setelah berhasil mengakses endpoint, permintaan tersebut akan diteruskan ke server sesuai bersamaan dengan data yang dikirimkan.

3. Server memberikan response ke API

Ketika menemukan data atau proses di server selesai, maka server akan menghubungi kembali API. Data yang dikembalikan umumnya berupa status code, message dan data.

4. API menyampaikan response ke aplikasi

Selanjutnya API akan meneruskan informasi dari server ke aplikasi.

5. RESTful API

RESTful API / REST API merupakan penerapan dari API (Application Programming Interface). Sedangkan REST (Representational State Transfer) adalah sebuah arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dimana metode ini sering diterapkan dalam pengembangan aplikasi. Dengan tujuannya untuk menjadikan sistem memiliki performa yang baik, cepat dan mudah untuk dikembangkan

(scale) terutama dalam pertukaran dan komunikasi data. API dikatakan RESTful jika memiliki fitur:

- a. Client – server : Client menangani front end dan server menangani back end dan keduanya dapat diganti secara independen satu sama lain.
- b. Stateless : Tidak ada data klien yang disimpan di server ketika ada permintaan dan status sesi disimpan di klien.
- c. Cacheable : Klien dapat men-cache respon (seperti browser yang men-cache elemen statis halaman web) untuk meningkatkan kinerja.

5.1. Komponen RESTful API

RESTful API memiliki 4 komponen penting yaitu:

- a. URL Design
RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut endpoint dalam pemanggilannya.

- b. HTTP Verbs

Setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di request client:

GET

GET adalah metode HTTP Request yang paling mudah, metode ini digunakan untuk membaca atau mendapatkan data dari sumber.

POST

POST adalah metode HTTP Request yang digunakan untuk membuat data baru dengan menyisipkan data dalam body saat request dilakukan.

PUT

PUT adalah metode HTTP Request yang biasanya digunakan untuk melakukan update data resource.

DELETE

DELETE adalah metode HTTP Request yang digunakan untuk menghapus suatu data pada resource.

- c. HTTP Response Code

HTTP Response Code adalah kode standarisasi dalam menginformasikan hasil request kepada client. Secara umum terdapat 3 kelompok yang biasa kita jumpai pada RESTful API yaitu :

2XX : adalah response code yang menampilkan bahwa request berhasil.

4XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi client.

5XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi server.

d. Format Response

Setiap request yang dilakukan client akan menerima data response dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data response tersebut barulah client bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.

5.2. Hal yang Perlu Diperhatikan Saat Membuat API

Untuk membuat REST yang baik ada beberapa hal yang perlu diperhatikan:

a. Endpoint gunakanlah Kata Benda Jamak (Plural)

/pets (mengembalikan list pet)

/pets/1 (mengembalikan pet dengan Id)

dan jangan menggunakan kata kerja

/getAllPets

/getAllPetsByCategory/1

b. Gunakan HTTP Method yang benar

Method-method standar yang biasa digunakan yaitu :

GET: Untuk mendapatkan data

POST: Untuk menambahkan data baru

PUT: Untuk memanipulasi Data

DELETE: Untuk menghapus Data

GET hanya untuk mengambil data, jangan gunakan untuk memanipulasi data

c. Gunakan sub-resource untuk sebuah relasi

Kadang kita mempunyai kebutuhan sebagai berikut, akan menampilkan semua data foods dari 1 pet, nah untuk kebutuhan sebagai berikut baiknya standarnya sebagai berikut

/pets/1/foods (mengeluarkan data foods dari data pets ID 1)

/pets/1/foods/2 (mengeluarkan data foods dengan ID 2)

d. Menggunakan Parameter

Untuk kebutuhan filtering bisa menggunakan parameter dengan standar berikut

/pets?category='cat' (mengeluarkan data pets dengan kategori 'cat')

- e. Menggunakan Paging
Khususnya dalam menggunakan method GET, untuk jumlah data yang besar kita bisa mengeluarkan data berdasarkan limit data (Paging)
/pets?limit=20 (mengeluarkan 20 data pets)

- f. Pengurutan Data
Disamping paging, kadang kita juga perlu untuk mengurutkan data berdasarkan request dari client

/pets?sort=updated_at (Mengeluarkan data berdasarkan dari yang update terbaru)

- g. HTTP Code
Berikut adalah List kode yang biasanya digunakan dalam REST

200: Untuk menunjukkan operasi yang di-request berhasil di proses

201 CREATED: Menunjukan operasi yang dilakukan berhasil membuat resource baru

400 BAD REQUEST: Biasanya kode error yang menunjukan dari sisi client misalkan client salah menginput data

403 FORBIDDEN: Response dari client misalnya client mengakses endpoint yang menyertakan Authorization tapi tidak mengirim kan token

401 NOT FOUND: Reponse yang menunjukan sumber yang direquest tidak tersedia

500 INTERNAL SERVER ERROR: terjadi kesalahan / error dari sisi server

- h. Gunakan Versioning
Penggunaan versi API sangatlah penting, apalagi di perusahaan yang banyak menggunakan versi API.

/v1/products

/v2/products

i. API Response

Tidak ada standart khusus mengenai API response namun umumnya sebagai berikut

- Success Response

```
1 {  
2   "status": "success",  
3   "data": [],  
4   "message": "Sukses mengambil data hewan peliharaan"  
5 }  
6
```

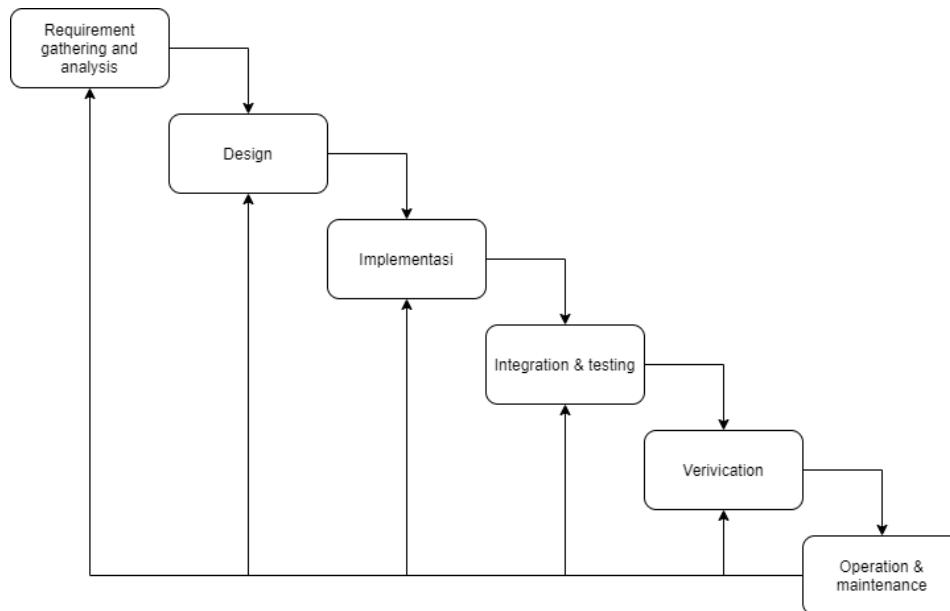
- Error Response

```
1 {  
2   "status": "error",  
3   "data": null,  
4   "message": "Gagal mengambil data hewan peliharaan",  
5   "errorCode": "500"  
6 }  
7
```

6. SDLC

Metode SDLC (Software Development Life Cycle) adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem rekayasa perangkat lunak. Metode SDLC hadir untuk membantu pengembangan produk. Terdapat banyak jenis SDLC, diantaranya:

a. Waterfall



Metode SDLC yang pertama adalah waterfall. Metode waterfall adalah metode kerja yang menekankan fase-fase yang berurutan dan sistematis. Disebut waterfall karena proses mengalir satu arah “ke bawah” seperti air terjun. Metode waterfall ini harus dilakukan secara berurutan sesuai dengan tahap yang ada.

Berikut adalah tahap-tahap pengembangan dalam metode waterfall.

1. Requirement gathering and analysis
Mengumpulkan kebutuhan secara lengkap untuk dianalisis dan mendefinisikan kebutuhan apa saja yang harus dicapai oleh program. Informasi dapat diperoleh melalui wawancara, diskusi, atau survey.
2. Design
Melakukan perancangan desain perangkat lunak sebagai perkiraan sebelum dibuatnya kode. Desain sistem dapat dibuat menggunakan Flowchart, Mind Map, atau Entity Relationship Diagram (ERD).
3. Implementasi
Implementasi ini adalah tahap dimana seluruh desain yang sebelumnya sudah dibuat diubah menjadi kode-kode program. Kode yang dihasilkan masih berbentuk modul-modul yang harus digabungkan di tahap selanjutnya.
4. Integration & testing
Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat sebelumnya dan melakukan pengujian untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desain dan fungsinya atau tidak.

5. Verification

Di tahap ini, pengguna atau klien yang langsung melakukan pengujian pada sistem, apakah sistem telah sesuai dengan yang disetujui atau belum sesuai.

6. Operation & maintenance

Tahap ini merupakan tahap terakhir dari model waterfall. Sistem yang sudah selesai dijalankan serta dilakukan pemeliharaan. Pemeliharaan berupa memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

Setiap metode yang digunakan pasti memiliki kelebihan serta kekurangannya tersendiri.

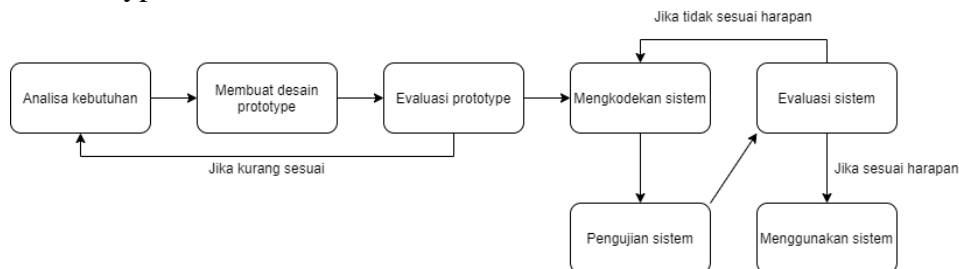
Berikut adalah kelebihan dari metode waterfall:

- Memiliki proses yang teratur, sehingga pengerjaan dapat terjadwal dengan baik dan mudah.
- Cocok untuk sistem dengan kompleksitas rendah (predictable).
- Setiap proses yang dilakukan tidak dapat saling tumpah tindih.

Berikut adalah kekurangan dari metode waterfall:

- Waktu pengerjaan relatif lebih lama, karena harus menunggu tahap sebelumnya selesai.
- Biaya yang dibutuhkan lebih mahal karena waktu pengembangan yang dibutuhkan lebih lama.
- Model waterfall ini kurang cocok untuk pengembangan proyek yang memiliki kompleksitas tinggi.

b. Prototype



Metode prototype adalah metode yang memungkinkan pengguna atau user memiliki gambaran awal tentang perangkat lunak yang akan dikembangkan, serta pengguna dapat melakukan pengujian di awal sebelum perangkat lunak dirilis.

Metode ini bertujuan untuk mengembangkan model menjadi perangkat lunak yang final. Artinya sistem akan dikembangkan lebih cepat dan biaya yang dikeluarkan lebih

rendah. Metode prototype ini memiliki tahap-tahap yang harus dilakukan dalam pengembangan perangkat lunak.

Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode prototype.

1. Analisa kebutuhan
Pada tahap ini pengembang melakukan identifikasi perangkat lunak dan semua kebutuhan sistem yang akan dibuat.
2. Membuat prototype
Membuat rancangan sementara yang berfokus pada alur program kepada pengguna.
3. Evaluasi prototype
Evaluasi dilakukan untuk mengetahui apakah model prototype sudah sesuai dengan harapan.
4. Mengkodekan sistem
Jika prototype disetujui maka akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Pengujian sistem
Setelah perangkat lunak sudah siap, perangkat lunak harus melewati pengujian. Pengujian ini biasanya dilakukan dengan White Box Testing, Black Box Testing, dan lain-lain.
6. Evaluasi sistem
Pengguna melakukan evaluasi apakah perangkat lunak sudah sesuai dengan apa yang diharapkan atau tidak. Jika ya, lakukan tahap selanjutnya. Jika tidak, ulangi tahap mengkodekan sistem dan pengujian sistem.
7. Menggunakan sistem
Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan.

Sebagai suatu metode yang sering digunakan, metode prototype pasti memiliki kelebihan dan kekurangan.

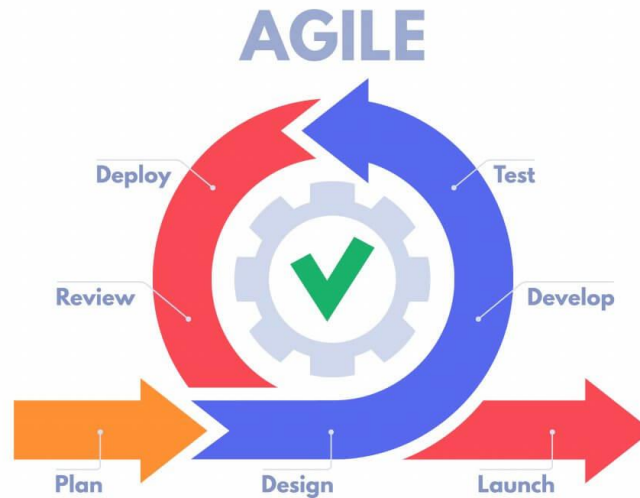
Berikut adalah kelebihan dari metode prototype:

- Mempersingkat waktu pengembangan perangkat lunak
- Penerapan fitur menjadi lebih mudah, karena pengembang mengetahui apa yang diharapkan

Berikut adalah kekurangan dari metode prototype:

- Proses yang dilakukan untuk analisis dan perancangan terlalu singkat
- Kurang fleksibel jika terjadi perubahan

c. Agile



Metode agile adalah metode yang fleksibel di mana pengembangan dilakukan dalam jangka pendek. Namun diperlukan adaptasi yang cepat dari developer terhadap perubahan dalam bentuk apa pun. Berikut merupakan tujuan dari agile, antara lain:

1. High – value & working app system
Menghasilkan produk dengan kualitas yang baik, dan memiliki nilai jual yang tinggi.
2. Iterative, incremental, evolutionary
Pengembangan dapat dilakukan secara iteratif, berulang-ulang, dan dapat mengalami perubahan jika diperlukan.
3. Cost control & value – driven development
Pengembangan perangkat lunak dapat sesuai dengan kebutuhan pengguna dan tim dapat dengan cepat merespon kebutuhan, sehingga waktu dan biaya pembuatan dari perangkat lunak dapat dikendalikan.
4. High – quality production
Kualitas dari perangkat lunak tetap terjaga, meskipun waktu dan biaya lebih sedikit.

5. Flexible & risk management
Meminimalisir terjadinya kesalahan pada program ataupun produk sebelum dilakukannya proses deploy aplikasi.
6. Collaboration
Kolaborasi ini dilakukan oleh setiap tim pengembang untuk mendiskusikan feedback yang diberikan oleh klien.
7. Self – organizing, self – managing teams
Pengembang diberikan akses untuk memanajemen sendiri urusan software development. Seorang manajer hanya bertugas sebagai penghubung antara pengembang dengan klien sehingga dapat mengurangi terjadinya miss communication.

Metode agile ini memiliki kelebihan dan kekurangan tersendiri.

Berikut adalah kelebihan dari metode agile:

- Perubahan dapat dengan cepat ditangani.
- Proses pengembangan perangkat lunak membutuhkan waktu yang relatif cepat dan tidak memerlukan sumber daya yang besar.
- Klien dapat memberikan feedback kepada pengembang dalam proses pembuatan program.

Berikut adalah kekurangan dari metode agile:

- Metode ini kurang sesuai dengan tim yang besar (lebih dari 20 orang).
- Tim harus selalu siap, karena perubahan dapat terjadi kapan saja.
- Metode ini kurang cocok untuk tim yang berkomitmen untuk menyelesaikan proyek bersama-sama.

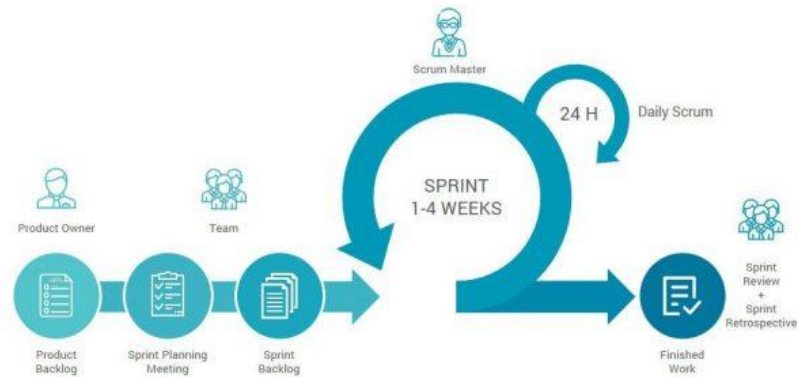
7. Scrum

Scrum adalah cara-cara yang dilakukan untuk memecahkan suatu masalah. Scrum sendiri menggunakan pendekatan dari metode lain yaitu Agile. Agile mengacu pada seperangkat metode dan praktik berdasarkan nilai dan prinsip yang diungkapkan dalam Agile Manifesto. Hal itu mencakup hal-hal seperti kolaborasi, pengorganisasian yang dilakukan sendiri oleh member tim, dan tim yang lintas fungsi.

Scrum adalah kerangka kerja yang digunakan untuk mengimplementasikan pengembangan Agile. Scrum pertama kali diperkenalkan dalam artikel Takeuchi and Nonaka yang berjudul “The New New Product Development Game” dan dipublikasikan oleh Harvard Business Review (HBR) pada tahun 1986. Metode Scrum digunakan pertama kali dalam pengembangan software dimulai oleh Jeff Sutherland, Easel Corporation pada tahun 1993. Selanjutnya, metode ini diformulasikan dan dipresentasikan pada Object Management Group tahun 1995 dengan judul paper “Scrum Development Process”.

Scrum dapat digunakan mengelola segala macam proyek mulai dari pembuatan software, website, hardware, marketing, hingga event planning. Scrum membantu tim untuk memecahkan masalah, dengan memiliki komunikasi yang kuat antar anggota tim tersebut.

7.1. Cara Kerja Metode Scrum



Untuk melaksanakan metode Scrum di tiap project yang dijalankan ada beberapa tahapan:

1. Menentukan tim
 Dalam mengerjakan suatu project, tentu kamu membutuhkan tim. Tahapan awal dalam memulai menerapkan metode Scrum adalah dengan menentukan anggota tim. Biasanya, anggota tim tak lebih dari 5-10 orang. Usahakan jumlah anggota seefektif dan seefisien mungkin.
2. Menentukan waktu pengerjaan
 Selanjutnya, dalam project management ada yang dikenal dengan sprint. Sprint sendiri serangkaian pekerjaan yang dilakukan untuk menyelesaikan suatu masalah khususnya penciptaan produk baru. Setelah menemukan tim, langkah menjalankan metode Scrum selanjutnya adalah menentukan waktu pengerjaan atau sprint. Biasanya, sprint berlangsung 7 hingga 30 hari.
3. Menentukan peran dalam tim
 Agar dapat bekerja maksimal, suatu tim haruslah memiliki struktur dan perannya masing-masing agar tidak tumpang tindih. Dalam project management, biasanya terdapat peran penting yang dimiliki oleh anggota tim. Peran ini adalah Scrum master, atau menurut Solstice bisa juga disebut sebagai project manager. Seorang Scrum master memastikan proyek berjalan dengan lancar dan sesuai. Peran selanjutnya dalam menjalankan metode Scrum adalah product owner, ia bertanggung jawab memastikan produk yang dihasilkan sesuai dengan kualitas yang diinginkan.

4. Mengumpulkan berbagai permasalahan
Langkah berikutnya adalah mengumpulkan berbagai hal yang didapat di lapangan. Dalam project management, hal ini disebut sebagai backlog. Berbagai permasalahan atau backlog ini kemudian dikumpulkan dan dibuat prioritas pengerjaannya.
5. Memulai sprint
Setelah semua langkah dijalankan, maka kamu bisa langsung memulai sprint tersebut. Dalam melaksanakan sprint, bisa saja ditemukan permasalahan atau backlog lain. Komunikasikan hal ini dengan product owner apakah backlog tersebut bisa dilaksanakan dalam sprint tersebut atau sprint selanjutnya.

7.2. Peranan dalam Metode Scrum

Scrum adalah salah satu metode terbaik yang dapat dimanfaatkan untuk keperluan manajemen proyek. Akan tetapi, supaya bisa digunakan secara maksimal, diperlukan tiga peranan khusus yang akan mengelola kinerja dalam kerangka kerja Scrum. Peran-peran ini juga penting untuk didefinisikan dan diperjelas, agar kerangka kerja Scrum dapat mensukseskan tim dan proyek. Tiga peran yang terdapat dalam Scrum

1. Product owner
Peranan utama yang terdapat dalam metode Scrum adalah product owner. Mereka bertugas untuk memastikan bahwa kinerja tim Scrum sudah selaras dengan tujuan keseluruhan proyek pengembangan yang ditargetkan badan usaha. Mereka harus memahami kebutuhan bisnis dari produk, seperti harapan para pelanggan, dan tren pasar. Karena mereka harus memahami bagaimana tim Scrum cocok dengan visi perusahaan, product owner biasanya akan berhubungan dengan manajer produk dan pemangku kepentingan lain di luar timnya.
2. Scrum master
Peranan berikutnya yang termasuk dalam kerangka kerja Scrum adalah scrum master. Mereka adalah para pekerja yang bertanggung jawab untuk memastikan bahwa tim proyek sudah beroperasi seefektif mungkin dengan nilai-nilai Scrum. Mereka harus menjaga tim agar tetap bekerja sesuai jalur dan metode Scrum, sambil merencanakan dan memimpin rapat, serta mengatasi hambatan dalam proyek. Scrum master juga biasanya akan dilimpahkan tugas yang lebih besar agar bisa memasukkan pola kerja Scrum pada seluruh perusahaan.
3. Development team
Peran terakhir yang terdapat dalam kerangka kerja metode Scrum adalah development team atau tim pengembangan. Tim ini terdiri dari para profesional yang melakukan pekerjaan secara langsung untuk menyelesaikan tugas dalam

sprint Scrum. Anggota development team sendiri termasuk computer engineer, desainer, penulis, data analyst, dan peran lain yang diperlukan untuk mencapai tujuan proyek. Tim pengembangan tidak hanya menunggu pesanan dari product owner. Mereka biasanya saling berkolaborasi untuk memetakan tujuan dan rencana untuk mencapai tujuan akhir proyek.

8. MVC

Model-View-Controller (MVC) adalah sebuah pola dalam desain aplikasi. MVC biasanya digunakan untuk mengimplementasikan antar muka pengguna (view), data (model) dan logika pengendali (controller). Beberapa pola design yang berdasarkan MVC adalah MVVM (Model-View-Viewmodel), MVP (Model-View-Presenter), dan MVW (Model-View-Whatever).

Tiga komponen dalam MVC adalah:

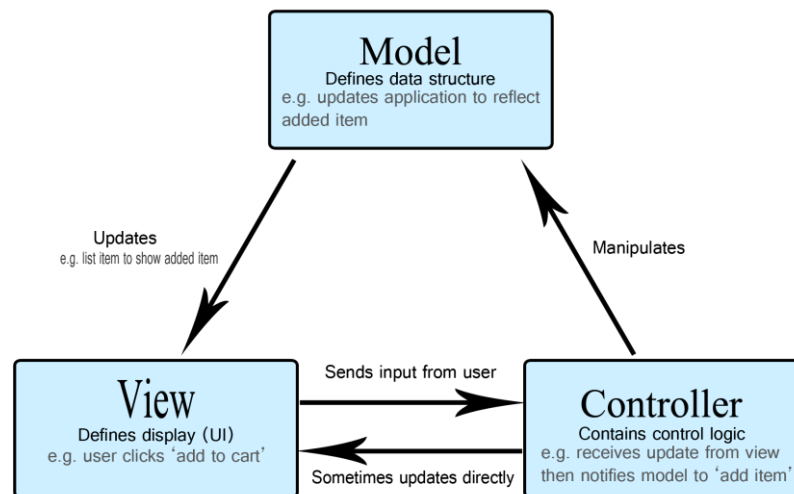
Model: Menangani data dan logika bisnis

View: Menangani layout dan tampilan

Controller: Mengarahkan perintah kepada model dan bagian view

Dengan konsep MVC ini aplikasi yang dibangun seakan memiliki bagian yang terpisah. Maka pengembang akan lebih cepat dan dapat fokus ke salah satu bagian saja. Banyak framework yang mendukung pola MVC, diantaranya Laravel, Codeigniter, Yii, dll

Alur kerja MVC



1. View akan request informasi untuk ditampilkan kepada controller.
2. Request yang diterima oleh controller akan diproses dan diteruskan ke model
3. Model akan mengolah dan mencari data di dalam database
4. Model akan memberikan informasi kepada controller untuk ditampilkan di view

5. Controller mengambil hasil olahan informasi oleh model dan menatanya di bagian view

Manfaat MVC

- Pengembangan aplikasi jadi lebih cepat dan efisien
- Penulisan kode jadi lebih rapi dan terstruktur
- Mempermudah proses testing
- Mempercepat proses perbaikan
- Mempermudah proses maintenance

9. Tech Stack

Tech stack adalah kombinasi dari teknologi yang digunakan untuk membangun dan menjalankan sebuah aplikasi atau proyek. Tech stack umumnya terdiri dari beberapa bahasa pemrograman, framework, database, peralatan frontend dan aplikasi yang terhubung lewat API. Tech stack dari sebuah aplikasi bisanya ditentukan dengan melihat beberapa aspek:

- Sesuai dengan tujuan bisnis aplikasi
- Efektif terhadap biaya pengembangan dan perawatan
- Memberikan pengalaman frontend tanpa kendala kepada pengguna
- Memberikan pengembang pengalaman pengembangan aplikasi yang cepat dan lancer
- Mudah dilakukan scale up jika diperlukan tanpa mengurangi performa aplikasi
- Mampu menyediakan keamanan untuk aplikasi

Terdapat banyak tech stack yang bisa digunakan sesuai kebutuhan aplikasi, namun diantaranya yang populer adalah:

a. MERN/ MEAN/ MEVN



- MongoDB (Database NoSQL)
- Express.js (Backend Framework)
- React/ Angular/ Vue (Frontend Framework)
- Node.js (Cross-platform Server)

Stack ini terdiri dari Mongo DB, Express JS dan Node JS sebagai inti dari stack. Untuk frontend stack ini bisa menggunakan salah satu dari React, Angular maupun Vue. Keuntungan dari stack ini adalah menggunakan satu bahasa saja untuk backend dan frontend (Javascript). Stack ini menyediakan pendekatan yang lebih praktis, cepat dan sangat efisien. Semua komponen saling berkomunikasi menggunakan JSON. Selain itu dengan banyaknya dukungan dari pustaka modul dapat menunjang pengembangan.

b. LAMP



- Linux (Sistem Operasi)
- Apache (HTTP Server)
- MySQL (Manajemen Database Relasional)
- PHP/Perl/Python (Bahasa Pemrograman)

Merupakan stack yang paling sering dijumpai dan bisa dikatakan sudah menjadi standart industri. Semua stack yang digunakan sifatnya open source sehingga pengembang dapat dengan mudah memodifikasi kode asal. Stack ini sebenarnya dapat berjalan di semua system operasi dan tidak hanya di linux. Untuk windows bisa menggunakan WAMP dan untuk MacOS menggunakan MAMP.

c. TALL



- Tailwind (CSS Framework)
- Alpine.js (Javascript framework)
- Laravel (Fullstack framework)
- Livewire (Laravel view component)

Merupakan stack untuk membuat aplikasi yang interaktif menggunakan Laravel. Bahasa pemrograman utama yang digunakan adalah PHP (Laravel) dan sedikit Javascript (Alpine). Dengan stack ini seorang pengembang PHP akan dengan mudah membuat aplikasi dengan cepat dan interaktif tanpa perlu memahami mendalam mengenai frontend framework javascript seperti React, Angular dan Vue.