

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian terdahulu

Rujukan penelitian yang kedua yaitu jurnal M. Miftakul Amin dosen Teknik Komputer Politeknik Negeri Sriwijaya Palembang pada tahun 2017 dengan judul Pengembangan Layanan Akses Nilai Akademik Berbasis *Web Services*. Dalam penelitiannya, peneliti mengembangkan aplikasi *web service* menggunakan bahasa pemrograman *PHP* dan didukung oleh *library nusoap* dan juga basis data *MySQL*. Sedangkan, pada sisi aplikasi *client* peneliti mengembangkan aplikasi menggunakan bahasa pemrograman *visual basic .Net* untuk mengonsumsi *web service*. Dengan, menggunakan *web service* maka pertukaran data antar aplikasi dapat dilakukan di *platform* yang heterogen.

Sedangkan aplikasi yang penulis buat, penulis menggunakan bahasa pemrograman *Node.JS* untuk server side. Didukung oleh *database server MongoDB* dan menggunakan notifikasi secara *real time* untuk mengetahui jadwal dosen yang sedang melaksanakan kegiatan belajar mengajar. Dan untuk client side, aplikasinya menggunakan teknologi *PWA*.

#### 2.2 Node.JS

*Javascript* merupakan bahasa pemrograman yang lengkap hanya saja selama ini di pakai sebagai bahasa untuk pengembangan aplikasi *web* yang berjalan pada sisi *client* atau *browser* saja. Tetapi sejak ditemukannya *Node.js* oleh Ryan Dhal pada tahun 2009, *Javascript* bisa digunakan sebagai bahasa pemrograman di sisi *server* sekelas dengan *PHP*, *ASP*, *C#*, *Ruby* dll dengan kata lain *Node.js* menyediakan *platform* untuk membuat aplikasi *Javascript* dapat dijalankan di sisi *server*.

Untuk mengeksekusi *Javascript* sebagai bahasa *server* diperlukan *engine* yang cepat dan mempunyai performansi yang bagus. *Engine Javascript* dari *Google* bernama *V8* yang dipakai oleh *Node.js* yang merupakan *engine* yang sama yang dipakai di *browser Google Chrome* (Pr, 2019).

### 2.3 Bootstrap

*Bootstrap* adalah sebuah *framework* yang dapat menyelesaikan permasalahan dalam mendesain *web*. Slogan dari *framework* ini adalah “*Sleek, intuitive, and powerful front-end framework for faster and easier web development*”, yang berarti kita dapat mendesain sebuah *website* dengan lebih rapi, cepat dan mudah. Selain itu *Bootstrap* juga *responsive* terhadap banyak *platform*, artinya tampilan halaman *website* yang menggunakan *Bootstrap* ini akan tampak tetap rapi, baik versi *mobile* maupun *desktop* (Widyantoro, 2009).

### 2.4 JQuery

*jQuery* sendiri adalah *Javascript Library* atau kumpulan kode/fungsi *Javascript* siap pakai, sehingga mempermudah kita untuk membuat kode *Javascript*. Atau dalam kesimpulannya *jQuery* menyederhanakan kode *Javascript*. Hal ini sesuai dengan slogannya “*Write less, do more*” (Muningmini, 2011).

### 2.5 NPM

NPM adalah tempat pendaftaran paket aplikasi / *library* untuk *nodejs*. Disana *developer* bisa mengupload kode mereka secara *public* / *open source* bisa digunakan untuk *developer* lain. Atau secara *private*. NPM juga menyediakan aplikasi dalam bentuk *CLI (Command Line Tools)* yang memudahkan para *developer* untuk menggunakan *install* paket yang berada pada *NPM software registry* ke dalam aplikasi mereka ([nodejs.org/en/knowledge/getting-started/npm/what-is-npm/](https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/), 2019).

### 2.6 GraphQL

*GraphQL* adalah bahasa *query* untuk *API* yang terletak pada *server*. *graphql* tidak terikat dengan beberapa jenis *database*. Sebaliknya, *graphql* didukung oleh kode dan data yang sudah ada. Sehingga akan membuat kode kita fleksibel, terutama pada saat mengakses data dari berbagai sumber ([graphql.org/learn/](https://graphql.org/learn/), 2019).

## 2.7 Socket.IO

*Socket.io* merupakan *library javascript* yang digunakan untuk membuat aplikasi *realtime*. Dan memungkinkan dua aplikasi yaitu, *server* dan *client* berkomunikasi satu sama lain. Kedua komponen tersebut menggunakan *API* / layanan yang sama untuk melakukan komunikasi. *Socket.io* juga mempunyai fitur *broadcasting*. Untuk mengirimkan suatu data ke banyak *client* (<http://socketio.blogspot.com/2017/02/what-is-socket-io.html>, 2019).

## 2.8 Prisma

*Prisma* merupakan *data layer* yang berfungsi sebagai *query engine* yang berjalan di sebuah server yang disebut dengan *prisma server*. *prisma server* akan menangani request yang datang dari *client* dan menterjemahkan request tersebut menjadi *database query*. sehingga, *API* dari *prisma server* akan dikonsumsi oleh *application server* dan memfokuskan *developer* untuk membangun aplikasi. Tanpa memperhatikan performa pada saat mengakses ke *database server*. *prisma* juga mendukung beberapa *database server* seperti *MySQL*, *MongoDB*, dan *PostgreSQL* ([www.prisma.io/docs/understand-prisma/prisma-introduction-what-why-how-j9ff/](http://www.prisma.io/docs/understand-prisma/prisma-introduction-what-why-how-j9ff/), 2019).

## 2.9 Arduino

*Arduino* adalah *platform* berbasis elektronik yang *hardware* dan *softwarena*nya mudah digunakan. Papan *arduino* bisa membaca *input* – lampu pada sensor, sidik jari pada sebuah tombo, atau bahkan pesan dari aplikasi *twitter*. Dan mengubahnya menjadi sebuah *output* – mengaktifkan sebuah motor, menyalakan lampu *LED*, mempublikasi suatu pesan / data secara *online*. Anda bisa memerintahkan papan *arduino* dengan cara mengirimkan beberapa perintah instruksi ke sebuah *microcontroller* di papan *arduino*.

Bertahun-tahun *arduino* telah menjadi cikal bakal dari ribuan proyek, dari alat / benda yang dijumpai sehari-hari sampai perangkat ilmiah yang sangat kompleks. Sebuah Komunitas dari seluruh dunia mulai dari siswa, seniman,

*programmer*, dan seorang profesional bahkan sekedar penggiat / hobi telah berkontribusi pada *platform arduino open source* ini. Kontribusi mereka telah membantu para pemula dan orang yang sudah mahir ([www.arduino.cc/en/guide/introduction](http://www.arduino.cc/en/guide/introduction), 2019).

### 2.10 MongoDB

*MongoDB* adalah *opensource database management system (DBMS)* yang menggunakan *model* yang berbasis *document* yang mendukung data dalam berbagai bentuk. *MongoDB* adalah salah satu *database nonrelational* yang ada di tengah tahun 2000-an dibawah naungan *database NoSQL* yang digunakan untuk aplikasi dengan data yang sangat banyak dan pemrosesan pekerjaan yang melibatkan data dimana data tersebut tidak cocok dengan *database relational*.

*MongoDB* tidak menggunakan istilah *table* dan *row* seperti di *database relational*, arsitektur *mongodb* terdiri dari *collection* dan *document* ([searchdatamanagement.techtarget.com/definition/MongoDB](http://searchdatamanagement.techtarget.com/definition/MongoDB), 2019).

### 2.11 Docker

Docker adalah sebuah *platform opensource* yang digunakan untuk membuat, *mendeploy* suatu aplikasi ke dalam sebuah *container*. *Container* mempermudah *developer* untuk mengemas sebuah aplikasi beserta bagian-bagian yang dibutuhkan, seperti *libraries*, kode aplikasi, *system libraries*, dan sebuah *runtime*. Di satu sisi, *docker* mirip seperti *virtual machine*. Akan tetapi, *docker* tidak perlu membuat sistem operasi *virtual*. Ini akan meningkatkan performa dari sebuah aplikasi dan mengurangi ukuran dari sebuah aplikasi tersebut ([opensource.com/resources/what-docker](http://opensource.com/resources/what-docker), 2019).

### 2.12 VPS

*VPS* adalah singkatan dari *virtual private server*, dengan *operating system* tersendiri dan *resource server* yang teralokasi, di dalam suatu server yang utuh (besar). Anda bisa menggunakannya ibarat anda mempunyai satu kepingan dari *hardware* fisik yang berfungsi sebagai *server* yang terpisah. Walaupun *server* fisik

terbagi, ada *element* terbagi secara individual di dalam server anda. *VPS* yang anda gunakan hanyalah milik anda, anda tidak akan membagi *CPU, RAM* dan data yang lainnya ke *server* lain. Dengan adanya *root access* terhadap *server vps*, anda bisa melakukan kustomisasi pada *server* anda tanpa mempengaruhi server lain. *Website* yang anda *hosting* akan berada di *isolated environment*. Sehingga, masalah yang terjadi pada *server* tetangga tidak akan berdampak pada *server* anda ([www.webhostingsecretrevealed.net/vps-hosting-guide/](http://www.webhostingsecretrevealed.net/vps-hosting-guide/), 2019).

### **2.13 Sistem Informasi Akademik**

Sistem informasi akademik adalah suatu sistem yang dibuat untuk mempermudah kegiatan administrasi akademik di kampus, yang mana kesemuanya diatur secara daring (online). Beberapa contoh kegiatan yang bersifat administratif di kampus adalah Penerimaan Mahasiswa Baru (PMB), penyusunan kurikulum dan jadwal kuliah, mengisi Kartu Rencana Studi (KRS), mengisi nilai (untuk dosen), mengelola data dosen, karyawan, dan mahasiswa, dsb ([ayokuliah.id/artikel/tips-kuliah/sistem-informasi-akademik-pengertian/](http://ayokuliah.id/artikel/tips-kuliah/sistem-informasi-akademik-pengertian/), 2019).

### **2.14 Perangkat Lunak Pendukung**

Dalam pembuatan aplikasi Sistem Informasi Akademik Teknik Komputer ini, penulis menggunakan beberapa macam perangkat lunak (software) pendukung, sebagai berikut :

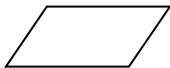
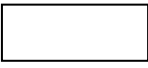


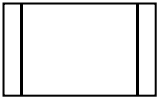
#### **2.14.1 Visual Studio Code**

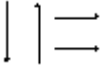
Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop Anda dan tersedia untuk Windows, macOS dan Linux. Muncul dengan dukungan bawaan untuk JavaScript, TypeScript dan Node.js dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti C ++, C #, Java, Python, PHP, Go) dan runtimes (seperti .NET dan Unity) ([code.visualstudio.com/docs](http://code.visualstudio.com/docs), 2019).

### 2.15 Pengertian Flowchart

Flowchart adalah suatu bagan yang menggambarkan arus logika dari data yang akan di proses dalam suatu program dari awal sampai akhir. Bagan alir program yang berguna bagi programmer untuk mempersiapkan program yang lebih kompleks. Bagan alir terdiri dari simbol yang mewakili fungsi-fungsi langkah program dan garis alir yang menunjukkan urutan dari simbol yang akan dikerjakan. Untuk melihat simbol *flowchart* bisa dilihat di Tabel 2.1 dibawah ini (Hartono, 2004).

**Tabel 2.1 Tabel Simbol Flowchart**

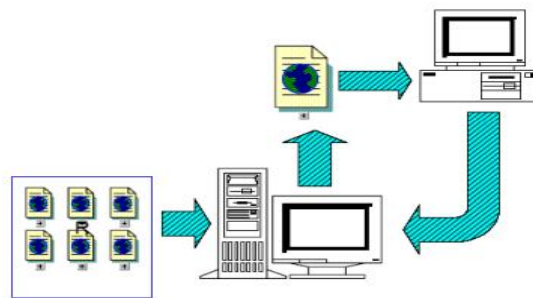
No.	Simbol	Nama	Fungsi
1		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
2		<i>Input/output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya.
3		<i>Proccess</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7		<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8		<i>Fanched Card</i>	Menyatakan input berasal dari kartu atau output data ke kartu.
9		<i>Document</i>	Mencetak keluaran dalm bentuk dokumen (melalui Printer)

10		<i>Flow</i>	Menyatakan jalannya arus suatu proses
----	---	-------------	---------------------------------------

(Sumber: Jogiyanto : 2004).

### 2.16 Pengertian Web Server

Web Server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima permintaan (request) berupa halaman web melalui HTTP atau HTTPS dari client yang dikenal dengan browser web dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk HTML. Standar arsitektur web bisa digambarkan seperti di Gambar 2.1 (Amin, 2019).



**Gambar 2.1** Standar Arsitektur Web

### 2.17 Pengertian Sistem

Pengertian sistem menurut beberapa ahli yaitu, Menurut (Sutabri, 2012) pada buku Analisis Sistem Informasi, pada dasarnya sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Selanjutnya menurut (McLeod, 2004) dalam buku Pengantar Sistem Informasi mendefinisikan sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan. Sistem juga merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan

tertentu. Sistem berasal dari Bahasa Yunani yaitu “systeme” yang mempunyai pengertian :

1. Suatu keseluruhan yang tersusun dari sekian banyak bagian.
2. Hubungan yang berlangsung diantara satuan-satuan atau komponen-komponen secara teratur.

Ciri-ciri atau sifat-sifat sistem adalah :

1. Terdiri dari komponen-komponen yang saling berinteraksi.
2. Mempunyai lingkungan luar.
3. Mempunyai interface (jalinan).
4. Terdiri dari masukan, pengolahan dan keluaran.

#### **2.18 Pengertian Informasi**

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. data merupakan bentuk jamak dari bentuk tunggal data atau item. Data adalah kenyataan yang menggambarkan sesuatu yang terjadi pada saat tertentu (Yakub, 2012) .

#### **2.19 Pengertian Sistem Informasi**

Sistem informasi adalah suatu komponen yang saling bekerja satu sama lain untuk mengumpulkan, mengolah, menyimpan dan juga menyebarkan informasi untuk mendukung kegiatan suatu organisasi, seperti pengambilan keputusan, koordinasi, pengendalian, analisis masalah, dan juga visualisasi dari organisasi. (Laudon dan Laudon, 2012). Menurut Simkin Mark G dalam bukunya yang berjudul “computer system for bussines”. Sistem informasi adalah sekumpulan eleme yang bekerja secara bersama-sama baik secara manual ataupun berbasis computer dalam melaksanakan pengolahan data yang berupa pengumpulan, penyimpanan, pemrosesan data untuk menghasilkan informasi yang bermakna dan berguna bagi proses pengambilan keputusan.



Menurut Burch dan Strater dalam bukunya “informasi system : theory and practice”. Sistem informasi adalah sekumpulan fungsi-fungsi yang bergabung secara formal dan sistematis yaitu :

1. Melaksanakan pengolahan data transaksi operasional.
2. Menghasilkan informasi untuk mendukung manajemen dalam melaksanakan aktifitas perencanaan, pengendalian dan pengambilan keputusan.
3. Menghasilkan berbagai laporan bagi kepentingan eksternal organisasi.

Dari definisi diatas terlihat bahwa sistem informasi merupakan satu kesatuan unsur (manusia dan peralatan) yang bekerja sama untuk melaksanakan pengolahan informasi dari mulai mengumpulkan, pengolahan, penyimpanan sampai pendistribusiannya.

## 2.20 Metode Pengembangan Sistem

### 2.20.1 Metode System Development Life Cycle (SLDC)

Metode ini adalah metode pengembangan sistem informasi yang pertama kali digunakan makanya disebut dengan metode tradisional. Metode ini *prototype* Adalah tahap-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi.

### 2.20.2 Metode waterfall

Merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara *linear*. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan.

### 2.20.3 Model Prototyping

Prototyping adalah proses *iterative* dalam pengembangan sistem dimana *requirement* diubah ke dalam sistem yang bekerja (*working system*)

yang secara terus menerus diperbaiki melalui kerjasama antara *user* dan analis. *Prototype* juga bisa dibangun melalui beberapa *tool* pengembangan untuk menyederhanakan proses.

#### **2.20.4 Model RAD (Rapid Application Development)**

RAD adalah penggabungan beberapa metode atau teknik terstruktur. RAD menggunakan metode *prototyping* dan teknik terstruktur lainnya untuk menentukan kebutuhan *user* dan perancangan sistem informasi selain itu RAD menekankan siklus perkembangan dalam waktu yang singkat (60 sampai 90 hari) dengan pendekatan konstruksi berbasis komponen.

#### **2.20.5 Model Spiral**

Model spiral pada awalnya diusulkan oleh Boehm, adalah *model* proses perangkat lunak *evolusioner* yang merangkai sifat iteratif dari *prototype* dengan cara kontrol dan aspek sistematis model sequensial linier. Model iteratif ditandai dengan tingkah laku yang memungkinkan pengembang mengembangkan versi perangkat lunak yang lebih lengkap secara bertahap.

#### **2.20.6 Object Oriented Technology**

*Object Oriented Technology* merupakan cara pengembangan perangkat lunak berdasarkan abstraksi objek-objek yang ada di dunia nyata. Dasar pembuatan adalah Objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Filosofi *Object Oriented* sangat luar biasa sepanjang siklus pengembangan perangkat lunak (perencanaan, analisis, perancangan dan implementasi) sehingga dapat diterapkan pada perancangan sistem secara umum: menyangkut perangkat lunak, perangkat keras dan *system* secara keseluruhan.

### 2.20.7 Model Functional Decomposition

Metodologi ini menekankan pada pemecahan dari sistem ke dalam subsistem-subsistem yang lebih kecil, sehingga akan lebih mudah untuk dipahami, dirancang dan ditetapkan.

Yang termasuk dalam kelompok metodologi ini adalah :

- HIPO (*Hierarchy plus Input Process Output*)
- *Stepwise Refinement* (SR) atau *Iterative Stepwise Refinement* (ISR)
- *Information Hiding*

### 2.20.8 Prescriptive Methodologies

Yang termasuk dalam metodologi ini adalah : ISDOS (*Information System Design dan Optimization System*), merupakan perangkat lunak yang dikembangkan di University of Michigan. Kegunaan dari ISDOS adalah mengotomatisasi proses pengembangan *system* informasi. ISDOS mempunyai dua komponen, yaitu :

1. PSL (*Program Statement Language*), merupakan komponen utama dari ISDOS, yaitu suatu bahasa untuk mencatat kebutuhan pemakai dalam bentuk *machine readable form*. PSL dirancang sehingga output yang dihasilkannya dapat dianalisis oleh PSA. PSL merupakan bahasa untuk menggambarkan sistemnya dan bukan merupakan bahasa pemrograman prosedural.
2. PSA (*Program Statement Analyzer*) merupakan paket perangkat lunak yang mirip dengan kamus *data* (*data dictionary*) dan digunakan untuk mengecek *data* yang dimasukkan, disimpan, dianalisis dan yang dihasilkan sebagai output laporan.

### 2.20.9 Model V

Model ini merupakan perluasan dari *model waterfall*. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam *model waterfall*. Jika dalam model *waterfall* proses dijalankan secara *linear*, maka dalam *model V* proses dilakukan bercabang.

#### **2.20.10 Metode End-user Development**

Disini pengembangan dilakukan langsung oleh *end-user*. Keterlibatan langsung *end-user* sangat menguntungkan, karena memahami benar bagaimana sistem bekerja. Artinya tahap analisis sistem dapat dilakukan lebih cepat. Kelemahan adalah pada pengendalian mutu dan kecenderungan tumbuhnya "*private*" sistem informasi dan Integrasi dengan sistem yang lain menjadi sulit (Saputra, 2013).