PROPOSAL

**IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK INTEGRASI DATA PERKEMBANGAN**

**COVID-19 DI SULAWESI SELATAN**



**BAMBANG IRIANTO**

**216 280 035**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH PAREPARE**

**2020**

**LEMBAR PERSETUJUAN**

**IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK INTEGRASI DATA PERKEMBANGAN**

**COVID-19 DI SULAWESI SELATAN**



**BAMBANG IRIANTO**

**216 280 035**

Setelah diperiksa / diteliti maka proposal ini dapat diajukan untuk mengikuti Seminar Proposal.

Disetujui Oleh :

Pembimbing I

**Syahirun Alam, ST.,MT**

**NBM. 883 314**

Pembimbing II

**Mughaffir Yunus, ST., MT.**

**NBM. 1252 809**

Mengetahui,

Ketua Program Studi Teknik Informatika

**Syahirun Alam, ST.,MT**

**NBM. 883 314**

# **PRAKATA**

**C:\Users\Administrator\Pictures\Bismill2.jpg**

Assalamu’alaikum Warahmatullah Wabarakatu.

Alhamdulillah, puji syukur kehadirat Allah SWT, yang Maha mendengar lagi Maha melihat dan atas segala limpahan rahmat, taufik dan hidayahNya sehingga penulis dapat menyelesaikan tugas kuliah proposal ini. Shalawat serta salam senantiasa tercurahkan kepada baginda Nabi Besar Muhammmad SAW beserta seluruh keluarga dan sahabatnya yang dengan setia berjuang menegakkan ajaran islam di muka bumi ini.

Adapun judul pada proposal ini **"Implementasi Web Service dengan Metode REST API untuk Integrasi Data Perkembangan COVID 19 di Sulawesi Selatan"** Dalam proposal ini penulis memohon maaf yang sebesar-besarnya apabila terdapat kekurangan dan kesalahan dalam penulisan dan artikulasi kata, kalimat maupun paragraf.

Penulis sepenuhnya menyadari bahwa penyelesaian Tugas ini tidak akan terwujud tanpa adanya dukungan dan bantuan semua pihak.

Akhirulkalam, semoga segala bantuan dari semua pihak yang telah membantu penulis serta harapan yang bagi penulis agar pembaca berkenan memberikan kritik dan saran yang membangun serta memperoleh manfaatnya. Aamiiin

Billahi Fii Sabilil Haq, Fastabiqul Khaerat

Wassalamu Alaikum Wr.Wb

Parapare, 10 Juli 2020

Bambang Irianto(216 280 035)

# **DAFTAR ISI**

SAMPUL

LEMBAR PERSETUJUAN

[PRAKATA i](#_Toc45876480)

[DAFTAR ISI ii](#_Toc45876481)

[DAFTAR TABEL iv](#_Toc45876482)

[DAFTAR GAMBAR v](#_Toc45876483)

BAB I [PENDAHULUAN 1](#_Toc45876485)

[A. Latar Belakang 1](#_Toc45876486)

[B. Rumusan Masalah 4](#_Toc45876487)

[C. Tujuan Penelitian 5](#_Toc45876488)

[D. Batasan Masalah 5](#_Toc45876489)

[E. Manfaat Penelitian 6](#_Toc45876490)

BAB II [TINJAUAN PUSTAKA 7](#_Toc45876492)

[A. Penelitian Terdahulu 7](#_Toc45876493)

[B. Website 9](#_Toc45876494)

[C. Web Service 9](#_Toc45876495)

[D. Application Programming Interface (API) 15](#_Toc45876496)

[E. Representational State Transfer (REST) 17](#_Toc45876497)

[F. JavaScript Object Notation (JSON) 19](#_Toc45876498)

[G. Postman 20](#_Toc45876499)

[H. Android 20](#_Toc45876500)

[I. Unified Modelling Language (UML) 21](#_Toc45876501)

[J. PHP 26](#_Toc45876502)

[K. Java 26](#_Toc45876503)

[L. XAMPP 27](#_Toc45876504)

[M. Server basis data MySQL 27](#_Toc45876505)

[N. CSS 28](#_Toc45876506)

[O. Bootstrap 28](#_Toc45876507)

[P. Android Studio 28](#_Toc45876508)

[Q. Visual Studio Code 29](#_Toc45876509)

[R. CodeIgniter 30](#_Toc45876510)

[S. CORONA VIRUS (COVID-19) 30](#_Toc45876511)

[T. Black Box 32](#_Toc45876512)

[U. Kerangka Pikir 35](#_Toc45876513)

[BAB III](#_Toc45876514) [Metode Penelitian 36](#_Toc45876515)

[A. Waktu dan Tempat Penelitian 36](#_Toc45876516)

[B. Jenis Penelitian 36](#_Toc45876517)

[C. Metode Pengumpulan Data 36](#_Toc45876518)

[D. Jenis Sumber Data 37](#_Toc45876519)

[E. Analisis kebutuan 38](#_Toc45876520)

[F. Desain Sistem 39](#_Toc45876521)

[DAFTAR PUSTAKA 40](#_Toc45876522)

# 

# **DAFTAR TABEL**

[Tabel 2. 1 Notasi Komponen Use Case Diagram 22](#_Toc45302859)

[Tabel 2. 2 Notasi Komponen Diagram Class Diagram 23](#_Toc45302860)

[Tabel 2. 3 Notasi Komponen Diagram Activity Diagram 25](#_Toc45302861)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Arsitektur Web Service 11](#_Toc45876942)

[Gambar 2. 2 Arsitektur Sistem Aplikasi 14](#_Toc45876943)

[Gambar 2. 3 Arsitektur API 16](#_Toc45876944)

[Gambar 3. 2 Sistem Yang Diusulkan 39](file:///F:\Proposal%20Bambang\revisi%20seminar%20pro\judulbaru.docx#_Toc46923011)

# **BAB I**

# **PENDAHULUAN**

1. **Latar Belakang**

Sistem komputer terdistribusi saat ini semakin penting dan banyak sekali digunakan mengingat dunia yang semakin didominasi oleh Teknologi Informasi. Hal tersebut salah satunya disebabkan oleh adanya standarisasi dalam arsitektur komputasi terdistribusi yang dinamakan dengan Web Service. Di beberapa literatur kita dapat temukan artikel-artikel ilmiah yang menyebutkan permasalahan arsitektur pada komputasi dan aplikasi terdistribusi dapat diselesaikan menggunakan web service, terutama di bidang pendidikan, kesehatan, bisnis dan lain-lain.

Dengan adanya web service, memungkinkan pengguna untuk dapat menghubungkan berbagai jenis software yang memiliki platform dan sistem operasi yang sama atau berbeda, sehingga pengguna tetap mendapatkan informasi dari website tanpa harus mengunjungi website yang memiliki platform dan sistem operasi berbeda.

Pada penerapan API (Application Programming Interface) bisa dikatakan sekumpulan teknik dasar yang meliputi : aturan, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API dapat menjelaskan cara sebuah tugas (task) tertentu dilakukan. Dalam pemrograman prosedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi/rutin yang disediakannya.

Pada konteks web, API merupakan pemanggilan fungsi lewat Hyper Text Transfer Protocol (HTTP) dan mendapatkan respon berupa Extensible Markup Language (XML) atau JavaScript Object Notation (JSON). Respon yang didapat dalam pemanggilan fungsi dapat bermacam-macam. Respon diatur oleh penyedia API.

Web API/web service ini dapat menggunakan arsitektur jaringan REST (representational state transfer), Istilah ini diperkenalkan pertama kali pada tahun 2000 pada disertasi doktoral Roy Fielding, salah seorang penulis utama spesifikasi HTTP. Istilah ini selanjutnya dipergunakan secara luas pada komunitas jaringan. REST secara spesifik merujuk pada suatu koleksi prinsip-prinsip arsitektur jaringan yang menggariskan pendefinisian dan pengalamatan sumber daya. Istilah ini sering digunakan dengan longgar untuk mendeskripsikan semua antarmuka sederhana yang menyampaikan data dalam domain spesifik melalui HTTP.

Penggunaan Konsep API web service ini dapat menguntungkan bagi pengguna untuk dapat lebih mudah dalam pengaksesan sebuah data, misalnya dengan adanya web service pada sebuah sistem website terdapat perubahan data, maka dengan tersistem data yang ada di aplikasi Android secara otomatis juga dapat berubah, dengan adanya API yang dibuat pengguna lain tidak dapat secara langsung mengakses database kita malainkan mereka harus request melalui internet lalu mengakses API kita dan kemudian direspon oleh API kita .

Wabah COVID-19 pertama kali dideteksi di Kota Wuhan, Provinsi Hubei, Tiongkok pada bulan Desember 2019, dan ditetapkan sebagai pandemi oleh Organisasi Kesehatan Dunia (WHO) pada 11 Maret 2020. Hingga 23 April 2020, lebih dari 2.000.000 kasus COVID-19 telah dilaporkan di lebih dari 210 negara dan wilayah, mengakibatkan lebih dari 195,755 orang meninggal dunia dan lebih dari 781,109 orang sembuh. (di akses pada "Coronavirus Update (Live): 307,627 Cases and 13,050 Deaths from COVID-19 Virus Outbreak - Worldometer". www.worldometers.info.)

Virus SARS-CoV-2 (COVID-19) diduga menyebar di antara orang-orang terutama melalui percikan pernapasan (droplet) yang dihasilkan selama batuk. Percikan ini juga dapat dihasilkan dari bersin dan pernapasan normal. Selain itu, virus dapat menyebar akibat menyentuh permukaan benda yang terkontaminasi dan kemudian menyentuh wajah seseorang. Penyakit COVID-19 paling menular saat orang yang menderitanya memiliki gejala, meskipun penyebaran mungkin saja terjadi sebelum gejala muncul.. Kasus positif COVID-19 di Indonesia pertama kali dideteksi pada 2 Maret 2020, ketika dua orang terkonfirmasi tertular dari seorang warga negara Jepang. Pada 9 April, pandemi sudah menyebar ke 34 provinsi dengan Jawa Timur, DKI Jakarta, dan Sulawesi Selatan sebagai provinsi paling terpapar.( Rothan, H. A. Dan Byrareddy, S. N, 2020).

Di Indonesia pada 06 juli 2020 sudah 64.958 orang yang positif Corona sedangkan di Sulawesi Selatan tercatat 5.974 orang yang positif Corona dan berstatus zona merah atau tidak aman (GIS Hub kawal covid 19: dikutip dari [https://gis-kawalcovid19.hub.arcgis.com/ [diakses](https://gis-kawalcovid19.hub.arcgis.com/%20%5bdiakses) 06 juli 2020] ).

Dalam situs penyedia data covid ini hanya dibuatkan API pada data dunia dan Indonesia saja tidak untuk berbagai provinsi di Indonesia. Dimana dari data yang diterima dari link kawal covid indonesia, setiap hari hampir ada kasus baru yang tercatat dari data perkembangan covid 19 di Provinsi khususnya di Sulawesi Selatan. Maka dari itu peneliti akan membuat sebuah web service data perkembangan covid 19 di Sulawesi Selatan yang dapat digunakan dan dimanfaatkan sebagai media informasi oleh semua kalangan.

Berdasarkan pemaparan diatas, pada skripsi ini akan dijelaskan mengenai bagaimana mengimplementasikan web service pada data perkembangan COVID 19 di Sulawesi Selatan menggunakan metode REST API. Maka penulis akan melakukan penelitian yang berjudul **"Implementasi Web Service dengan Metode REST API untuk Integrasi Data Perkembangan COVID 19 di Sulawesi Selatan"**

1. **Rumusan Masalah**

Berdasarkan latar belakang permasalahan yang telah dijelaskan di atas, maka penulis membuat perumusan masalah yang menjadi fokus dari penelitian sebagai berikut:

Bagaimana mengimplementasikan web service dengan metode REST API untuk melakukan integrasi data pada Perkembangan COVID 19 di Sulawesi Selatan yang terdiri dari beberapa domain website dan database, sehingga data dapat diakses secara terpusat.

1. **Tujuan Penelitian**

Tujuan penelitian berdasarkan latar belakang dan perumusan masalah di atas adalah mengimplementasikan web service dengan metode REST API dalam membuat suatu sistem yang berguna untuk integrasi data agar dapat diakses secara terpusat.

1. **Batasan Masalah**

Dalam menyusun penelitian untuk tugas akhir penulis memberi batasan masalah, sehingga pembahasan dalam tugas akhir ini tidak terlalu menyimpang pada tujuan penelitian semula. Berdasarkan perumusan masalah diatas dapat ditarik batasan masalah sebagai berikut:

1. Data yang akan digunakan adalah data pada Tanggap Covid di Sulawesi Selatan.
2. Teknologi yang akan digunakan adalah teknologi Service Oriented Architecture (SOA) atau lebih dikenal dengan sebutan Web Service.
3. Metode yang akan digunakan adalah Web Service dengan metode REST/RESTfull API.
4. **Manfaat Penelitian**

Manfaat yang diharapkan dengan adanya penelitian tugas akhir ini, sebagai berikut:

1. Manfaat bagi penulis
   1. Mampu mengimplementasikan pengetahuannya mengenai web service yang didapat selama berada di bangku perkuliahan.
   2. Dapat mengetahui lebih jauh mengenai penerapan metode REST API pada web service.
2. Manfaat bagi perguruan tinggi
   1. Menambah koleksi penelitian yang terkait dengan kemajuan teknologi dan informasi.
   2. Meningkatkan jumlah peminat penelitian dikalangan mahasiswa.
3. Manfaat bagi pembaca

Dapat dijadikan bahan referensi penelitian guna penelitian selanjutnya beserta pengembangannya.

# 

# **BAB II**

# **TINJAUAN PUSTAKA**

1. **Penelitian Terdahulu**
2. Rancang Bangun Web Service Perpustakaan Universitas Sam Ratulangi (2014).

Bramwell A. Kasaedja, Rizal Sengkey, ST,. MT dan Oktavian A. Lantang, ST., MT (B. A. Kasaedja, 2014) pada penelitian mereka yang memiliki judul "Rancang Bangun *Web Service* Perpustakaan Universitas Sam Ratulangi" melakukan implementasi terhadap teknologi *web service* dengan menggunakan arsitektur SOAP (*Simple Object Access Protocol*) dan *output* berupa XML WSDL. Penelitian ini memiliki tujuan melakukan perancangan dan mengembangkan sebuah sistem yang terintegrasi yang dibangun dengan menggunakan *web service* untuk pengelolaan anggota perpustakaan, pengelolaan peminjaman dan pengelolaan katalog buku. Hasil yang didapatkan adalah sebuah *web service* yang telah terhubung dengan MySql. Ketika *testing* dilakukan maka browser akan menampilkan SOAP *Request* dan SOAP *Response*. *Method* yang dipanggil akan melakukan respon dengan menampilkan data sesuai dengan yang telah didefinisikan di dalam *class*. Kesimpulan yang didapat berdasarkan penelitian tersebut adalah *web service* sebagai penyedia *service* memiliki fungsi-fungsi yang dapat mendukung dalam proses pengelolaan katalog, data peminjam dan keanggotaan perpustakaan sehingga menjadi lebih mudah. Selain itu, informasi-informasi perpustakaan dapat diakses dengan lebih mudah oleh siapa saja dan dimana saja melalui *web client*.

1. *Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment. (2014).*

Anil Dudhe dan S.S. Sherekar, Ph.D (A. Dudhe and S. Sherekar, 2014) pada penelitian mereka yang berjudul "*Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment*". Mereka melakukan perbandingan dan pengujian *performance* antara RESTful *web service* dan SOAP *web service* dengan mengembangkan *web service client* pada perangkat mobile. Mereka mengimplementasikan RESTful dan SOAP *web service*, serta melakukan *deploy* di Apache Tomcat web server serta Google App Engine. Kesimpulan dari penelitian tersebut adalah REST *services* memiliki proses lebih cepat (membutuhkan waktu yang lebih sedikit) daripada SOAP *web services* ketika mereka melakukan pengujian pada localhost dengan menggunakan Apache Tomcat Web Server. Selain itu mereka juga membuktikan bahwa REST memiliki *performance* yang lebih baik daripada SOAP pada *Cloud server Google App Engine*.

1. **Website**

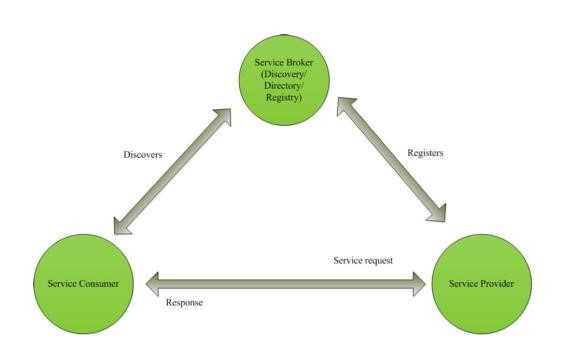
Website dapat diartikan sebagai kumpulan dari halaman-halaman yang mampu menampilkan informasi berupa data gambar diam ataupun bergerak, data teks, data animasi, video, suara dan atau gabungan dari keseluruhannya. Websitememiliki sifat statis maupun dinamis serta membentuk suatu rangkaian yang saling terkait satu sama lain dimana rangkaian tersebut dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Dapat dikatakan memiliki sifat dinamis apabila informasi dari suatu website dapat berubah-ubah dan berisi informasi yang interaktif secara dua arah yang berasal dari pemilik dan pengguna website. Suatu website dapat dikatakan statis apabila konten di dalam website tidak berubah dan isi informasi yang dimiliki searah berasal dari pemilik website (D. A. R. Anggiani, R. Eko, 2012).

1. **Web Service**

Menurut W3C, Web service adalah suatu sistem perangkat lunak yang didisain untuk mendukung interaksi mesin ke mesin pada suatu jaringan. Ia mempunyai suatu interface yang diuraikan dalam suatu format machine-processible seperti WSDL. Sistem lain yang berinteraksi dengan Web service dilakukan melalui interface/antar muka menggunakan pesan seperti pada SOAP. Pada umumnya pesan ini melalui HTTP dan XML yang merupakan salah satu standard web. Perangkat Lunak aplikasi yang ditulis dalam berbagai bahasa pemrograman dan berjalan pada berbagai platform dapat menggunakan Web Service untuk pertukaran data pada jaringan komputer seperti Internet dalam cara yang serupa dengan komunikasi interprocess pada komputer tunggal. Interoperabilitas ini (sebagai contoh, antara Java dan Python, atau Microsoft Windows dan aplikasi Linux) adalah dalam kaitan dengan penggunaan dari open standard.

Sedangkan menurut Michael C. Daconta, Web Service adalah aplikasi perangkat lunak yang dapat ditemukan, diuraikan, dan diakses berdasarkan pada XML dan protokol standard Web pada intranet, extranet, dan Internet. Web Service adalah aplikasi perangkat lunak yang tersedia pada Web yang melaksanakan fungsi yang spesifik. Berikutnya, kita akan lihat di pertengahan dari definisi yakni “ditemukan, diuraikan, dan diakses berdasarkan pada XML dan protokol standard Web.” Dibangun pada XML, suatu standard yang didukung dan diterima oleh beribu-ribu vendor di seluruh dunia, Web Service pertama fokus pada interoperabilitas. XML adalah sintaksis dari pesan, dan Hypertext Transport Protocol (HTTP), bagaimana aplikasi mengirimkan pesan XML ke Web Service dalam rangka berkomunikasi.

Menurut W3C, arsitektur dasar teknologi *web service* adalah *Service Oriented Architecture* (SOA) dan terdapat tiga komponen yang membentuk *web service* :



Gambar 2. 1 Arsitektur Web Service

* 1. *Service provider* : Komponen ini menyediakan layanan layanan dan melakukan pengelolaan terhadap sebuah *registry* yang menjadikan layanan tersebut tersedia.
  2. *Service broker* : Komponen ini merupakan tempat terjadinya transaksi untuk layanan. *Service brokers* memiliki peranan sebagai penghubung antara penyedia layanan dan pemohon layanan.
  3. *Service requestor* : Komponen ini bekerja dengan *service brokers* untuk menelusuri layanan pada web, kemudian memanggil layanan tersebut untuk membuat aplikasi.

Selain itu terdapat tiga operasi di dalam *web service* (D. Tidwell, 2000):

1. *Publish / Unpublish* : *Publish* dan *unpublish* melibatkan *advertising service* ke *registry* (disebut *publishing*) atau menghapus entri-entri (disebut *unpublishing*).
2. *Find* : Operasi *find* dilakukan oleh *service requestor* dan *service* *broker* secara bersama-sama. *Service requestor* menggambarkan jenis layanan yang dicari oleh *user*, dan *service broker* menampilkan atau memberikan hasil yang terbaik sesuai permintaan.
3. *Bind* : Operasi *bind* berlangsung di antara *service requestor* dan *service provider*. Kedua belah pihak melakukan negosiasi yang sesuai agar pemohon dapat mengakses dan memanggil layanan dari penyedia.

*Web service* dibangun dengan 5 (lima) standar dasar (D. Tidwell, 2000):

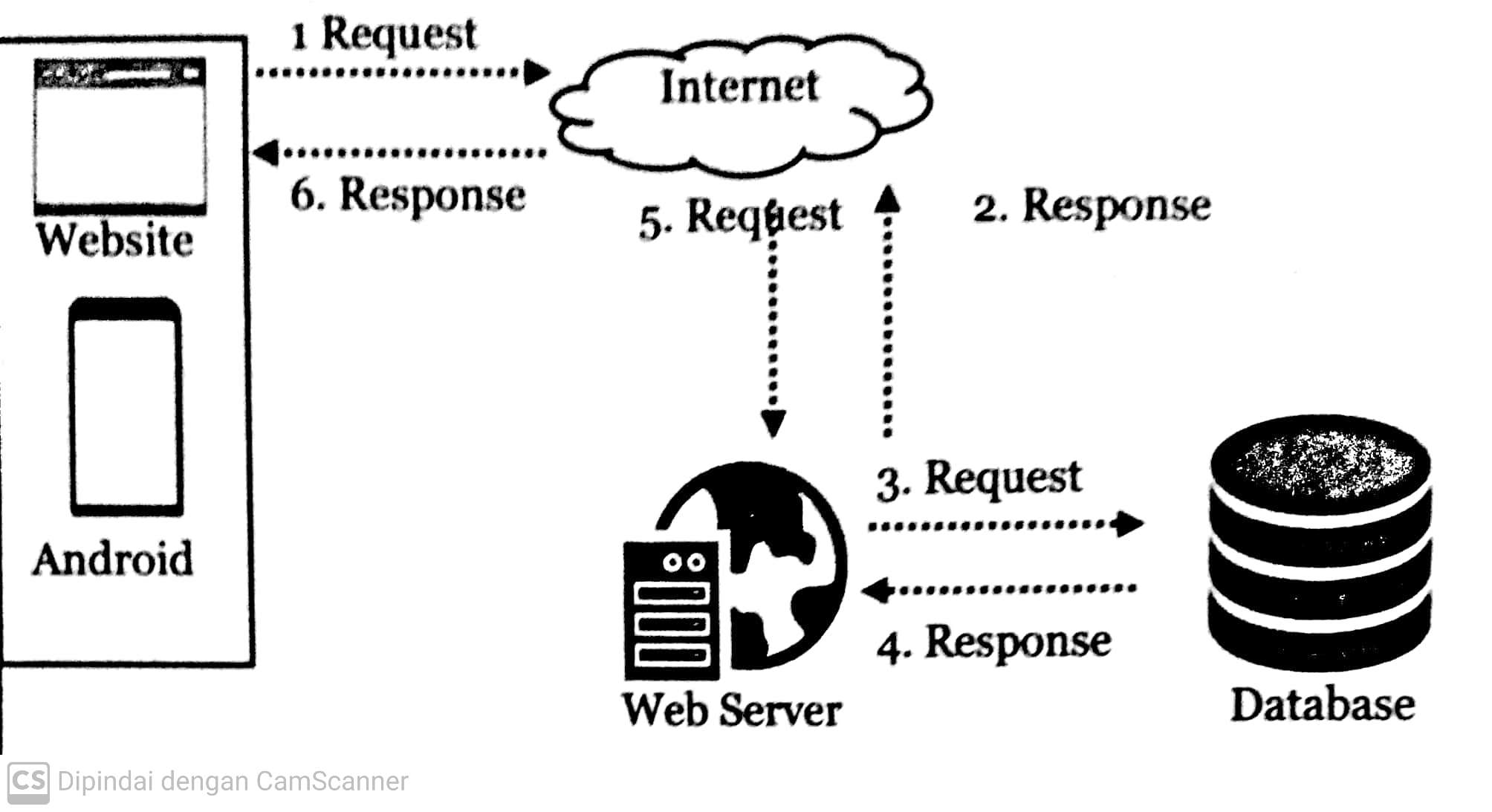
1. *XML* : *Extensible Markup Language* merupakan suatu format standar yang biasa digunakan di dalam proses pertukaran data.
2. *WSDL* : *Web Services Definition Language* adalah standar di dalam *web service* yang menyediakan aturanaturan *object* dan fungsi.
3. *HTTP(S)* : *Hypertext Transfer Protocol (Secure)* merupakan protokol dalam pertukaran data yang berbasis text dan saat ini telah menjadi protokol standar di dalam pengaksesan internet melalui browser dan media lainnya.
4. *UDDI* : *Universal Description, Discovery, and Integration* adalah suatu penampung *registry* *web services* yang saat ini telah dikembangkan agar dapat diakses dan diintegrasikan.
5. *SOAP* : *Simple Object Application Protocol* yaitu protocol yang mengemas *object* dan fungsi pada saat proses pertukaran data.

Web service juga dapat disebut sebagai salah satu konsep sistem perangkat lunak yang dimodel untuk mendukung interaksi mesin kemesin melalui jaringan. Web service memiliki pemodelan yang dideskripsikan dalam format yang dapat dibaca oleh mesin (Sibagariang, 2016).

Kelebihan yang dimiliki oleh *web service* (E. Sutanta dan K. Mustofa, 2012), yaitu:

* 1. *Language independent*, dapat diakses dan dibangun oleh bahasa pemrograman apapun.
  2. Lintas *platform*, tetap memungkinkan terjadinya pertukaran data, walaupun menggunakan perangkat-perangkat dengan sistem operasi yang berbeda.
  3. Jembatan penghubung dengan database, web service dapat diibaratkan sebagai “jembatan” penghubung antara database dengan aplikasi tanpa memerlukan adanya driver database. Selain itu, juga tanpa perlu mengetahui database yang digunakan oleh server dan bagaimana bentuk struktur database tersebut apabila ingin mengaksesnya. Aplikasi hanya cukup mengetahui berbagai macam method dan fungsi yang telah disediakan oleh web service, sehingga dapat melakukan pemanfaatan terhadap fasilitasnya.
  4. Penggunaan kembali komponen aplikasi, aplikasi-aplikasi yang berbeda dapat menggunakan sebuah fungsi yang sama.
  5. Proses pertukaran data menjadi semakin lebih mudah dan cepat tanpa harus menyesuaikan aplikasi, database, dan platform yang digunakan.

Arsitektur sistem aplikasi yang dirancang dapat menggambarkan konsep atau alur kerja aplikasi da fitur-fitur dalam aplikasi. Arstektur sistem aplikasi dapat dilihat pada gambar berikut : (Eko dan Angga, 2019).

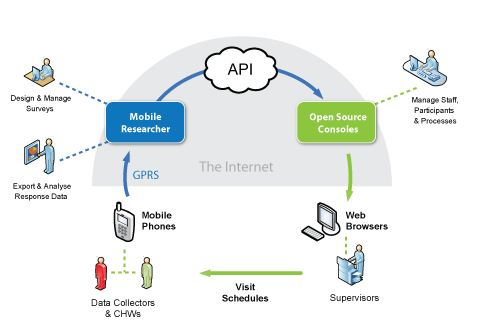


Gambar 2. 2 Arsitektur Sistem Aplikasi

Penjelasan Gambar 2.2 :

1. Request data ke webserver dilakukan oleh pengguna website dan android melalui internet
2. Request dari pengguna website dan android ke webserver diteruskan oleh internet
3. Request diterima oleh webserver melalui pengguna website dan android, selanjutnya request oleh webserver berupa data di database sesuai request pengguna website dan android.
4. Request diterima oleh database dan membalasnya ke webserver sesuai request.
5. Data dikirim sesuai request pengguna website dan android oleh webserver melalui internet.
6. Respon diteruskan internet ke pengguna website dan android dari webserver.
7. **Application Programming Interface (API)**

API atau kepanjangan dari (Application Programming Interface) adalah sebuah metode yang digunakan oleh sebuah aplikasi untuk berinteraksi dengan sistem operasi, protokol komunikasi, atau program lain *RESTful API* merupakan implementasi dari *API. RESTful* itu protokol/aturan untuk melakukan *REST*. Jadi *RESTful* itu udah pasti *REST*, namun *REST* belum tentu bisa disebut *RESTful*. Walaupun terlihat sama tetapi mereka berbeda. (Abidin, 2015).



Gambar 2. 3 Arsitektur API

API dapat menjelaskan cara sebuah tugas (task) tertentu dilakukan. Dalam pemrograman procedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi/rutin yang disediakannya. API menyediakan fungsi dan perintah dengan bahasa yang lebih terstruktur dan lebih mudah untuk dipahami oleh programer bila dibandingkan dengan System Calls, hal ini penting untuk aspek editing dan pengembangan, sehingga programer dapat mengembangkan sistem dengan mudah. API juga dapat digunakan pada Sistem Operasi mana saja asalkan sudah ada paket-paket API nya. (Abidin, 2015).

1. **Representational State Transfer (REST)**

Salah satu metode yang dapat digunakan untuk pengembangan teknologi *web service* adalah REST (*Representational State Transfer*). Metode REST *web service* menerapkan konsep perpindahan antar *state*. *State* yang dimaksud disini dapat digambarkan apabila *browser* melakukan permintaan suatu *web*, maka *server* akan melakukan pengiriman *state* halaman *web* yang sekarang ke *browser* . Ide dasar dari metode REST adalah menggunakan mekanisme HTTP untuk menghubungkan aplikasi dibandingkan dengan menggunakan mekanisme yang kompleks, seperti CORBA, RPC dan SOAP. REST berfokus utama pada interaksi sumber daya dan mengubah *state*, bukan berfokus pada mengirim dan menerima pesan seperti pada *web service* berbasis SOAP.( R. Ramanathan and T. Korte, 2014)

REST didefinisikan sebagai seperangkat prinsip arsitektur yang digunakan untuk membangun *web service* yang berfokus pada sumber daya sistem, termasuk bagaimana sumber daya *states* ditujukan dan ditransfer melalui HTTP oleh berbagai klien yang ditulis ke dalam bahasa-bahasa yang berbeda. Selain itu, REST bernavigasi melalui *link-link* HTTP untuk melakukan aktivitasaktivitas tertentu, sehingga seakan-akan terjadi perpindahan *state* satu sama lain.( A. Rodriguez dan I. B. M, 2008).

Metode REST didasari oleh empat prinsip (R. Ramanathan dan T. Korte, 2014):

1. Identifikasi *resources* melalui *Uniform Resource Identifier (URI)*: REST *web service* menyediakan sejumlah *resource* yang mengidentifikasikan tujuan dari interaksi dengan para *client*. *Resource* diidentifikasi oleh URI yang dapat mengakomodasi ruang pengalamatan untuk *resource* dan *service discovery*.
2. Keseragaman di dalam *resource*: Setiap *resource* di dalam REST dimanipulasi oleh empat macam operasi, yaitu GET, POST, PUT dan DELETE untuk membaca, memperbarui, membuat, dan menghapus.
3. Penggunaan *hyperlink* untuk interaksi *stateful*: Semua interaksi dengan *resource* bersifat *stateless*. Interaksi *stateful* adalah salah satu di mana adanya transfer *state* secara eksplisit. Ada sejumlah cara yang dapat dilakukan untuk melakukan pertukaran *state*, seperti menulis ulang URI. *State* juga dapat ditanamkan di dalam pesan respon, sehingga dapat digunakan untuk referensi kembali di waktu mendatang.
4. *Message* dengan *self-description*: *resource* pada REST tidak terikat sehingga dapat melakukan akses terhadap berbagai macam konten dengan format PDF, JSON, HTML, XML, JPEG, *plain text* dan lainnya.

REST didasarkan pada operasi-operasi yang bersifat universal dan dapat digunakan untuk berbagai macam penyimpanan data dan sistem pencarian. Operasi-operasi ini biasa disebut dengan singkatan CRUD (*Create, Read, Update, Delete*). (M. Philip dan G. Clynch, 2013).

1. **JavaScript Object Notation (JSON)**

JSON dirancang untuk menjadi bahasa pertukaran data yang dapat dibaca manusia, serta mudah untuk diuraikan dan digunakan oleh komputer . JSON sangat *support* JavaScript dan paling cocok digunakan untuk aplikasi JavaScript, sehingga memberikan keuntungan kinerja yang signifikan dibandingkan dengan XML, yang masih membutuhkan *library* tambahan untuk mengambil data dari *Document Object Model* (DOM). Selain itu, JSON dapat bekerja seratus kali lebih cepat dibandingkan dengan XML pada *modern browser* (N. Nurseitov dkk, 2009). JSON dibangun atas dua struktur (dikutip pada “http://www.json.org/.”):

1. Sekumpulan pasangan nilai/nama. Dalam berbagai bahasa, direalisasikan sebagai sebuah *record*, objek, *struct*, tabel *hash*, kamus, *array* asosiatif atau *keyed list*.
2. *Ordered list* dari sekumpulan nilai-nilai. Dalam kebanyakan bahasa, ini direalisasikan sebagai *vector*, *array*, *list* atau urutan.

Keuntungan dari menggunakan JSON sebagai *message format* (R. Ramanathan dan T. Korte, 2014) :

1. JSON adalah format pertukaran data, sedangkan XML adalah format pertukaran dokumen.
2. JSON mudah untuk dibaca dan ditulis. Proses yang terjadi pada JSON pun sederhana, hal itu dikarenakan strukturnya.
3. JSON ringan, *easy to handle and parse* jika dibandingkan dengan XML.
4. **Postman**

*Postman* merupakan sebuah *software* yang memuat fungsi lengkap pengembangan sistem dalam mengirimkan dan menerima respon *server*. *Software* ini mendukung pengembangan sistem REST API dengan mengklasifikasi *request* berdasarkan *request method*, URL dan parameter-parameter *request* (Postdot, 2017).

1. **Android**

Menurut Safaat (2012), android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android menyediakan platform yang memberikan kebebasan kepada pengguna untuk membuat aplikasi mereka sendiri. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh Google pada tahun 2005.

Tampilan Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Sejak pertama diluncurkan android sudah memiliki fiturnya masing-masing berdasarkan dengan versinya. Setiap kemunculan versi android yang baru selalu disertai dengan nama yang unik sesuai dengan urutan alfabet seperti Cupcake, Donut, Eclair, Froyo, Ginggerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat, Oreo dan yang terakhir Pay

1. **Unified Modelling Language (UML)**

*Unified Modelling Language* (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar, 2005). UML merupakan kesatuan dari bahasa pemodelan yang dikembagkan oleh *Booch, Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode *Booch* dari *Grady Booch* sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design kedalam empat tahapan iteratif, yaitu : identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi.

1. *Use Case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Dharwiyanti, 2003). Berikut ini ada beberapa notasi komponen use case diagram yang terlihat pada Tabel 2.1.

Tabel 2. 1 Notasi Komponen Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 | |  | | --- | |  | | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

1. *Class Diagram*

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain beroientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/ fungsi).*Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain. Berikut ini ada beberapa komponen diagram class diagram yang terlihat pada Tabel 2.2.

Tabel 2. 2 Notasi Komponen Diagram Class Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung  padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

Hubungan Antar *Class*:

1. *Asosiasi*, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui *eksistensi class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. *Agregasi*, yaitu hubungan yang menyatakan bagian (terdiri atas).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan(*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.
5. *Activity Diagram*

*Activity* diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses *paralel* yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan *state* diagram khusus, dimana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya(*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Berikut ini ada beberapa notasi komponen diagram activity diagram yang terlihat pada Tabel 2.3.

Tabel 2. 3 Notasi Komponen Diagram Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

1. **PHP**

PHP adalah bahasa scripting yang menyatu dengn HTML dan dijalankan pada serverside. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirim hanya hasilnya saja. PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script serverside* dalam pengembangan *web* yang disisipkan pada dokumen HTML. PHP merupakan *software Open-Source* yang disebarkan dan dilisensi secara gratis serta dapat di *download* secara bebas dari situs resminya. (Wulandari, 2017.).

1. **Java**

Java adalah bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

1. **XAMPP**

*XAMPP* adalah perangkat lunak bebas, yang menggabungkan 3 perangkat lunak yaitu, *apache*, phpmyadmin dan MySQL ke dalam satu paket. *XAMPP* mempunyai kelebihan untuk bisa berperan sebagai *web server* dengan menggunakan *apache* untuk simulasi pengembangan *website*, sehingga bagi pengembang *website* dapat mempresentasikan ke pihak lain secara langsung dari komputer, tanpa harus terkoneksi ke *internet*.

*XAMPP* juga dilengkapi dengan fitur manajemen basis data phpmyadmin sehingga para pengembang *website* dapat mengembangkan *website* berbasis data secara mudah. (Noor, 2016).

1. **Server basis data MySQL**

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi, yaitu Free Software dan Shareware. MySQL yang Free Software bebas digunakan untuk keperluan pribadi atau usaha tanpa harus membayar atau membeli lisensi GNU/GPL (General Public Lisence).

1. **CSS**

CSS adalah salah satu bahasa pemrograman desain web (style sheet languages) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan bahasa penanda (markup language). Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL.

1. **Bootstrap**

Bootstrap merupakan produk open source yang dibuat oleh Mark Otto dan Jacob Thornton yang ketika awal dirilis, keduanya merupakan karyawan di twitter, dan ada kebutuhan untuk menstandarisasi perlengkapan (toolsets) dari antarmuka para insinyur yang ada di perusahaan.

1. **Android Studio**

Android Studio adalah Lingkungan Pengembangan Terpadu (Integrated Development Environment/IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android

1. **Visual Studio Code**

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst).

Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor-teks editor yang lain.

Teks editor VS Code juga bersifat open source, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Kode sumber dari VS Code ini pun dapat dilihat di link Github. Hal ini juga yang membuat VS Code menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan VS Code ke depannya.

1. **CodeIgniter**

CodeIgniter merupakan aplikasi sumber terbuka yang berupa kerangka kerja PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan pengembang web untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.6.

Kerangka kerja secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programer, tanpa harus membuat fungsi atau class dari awal.

1. **CORONA VIRUS (COVID-19)**

Penyakit koronavirus 2019 (bahasa Inggris: coronavirus disease 2019, disingkat COVID-19) adalah penyakit menular yang disebabkan oleh SARS-CoV-2, salah satu jenis koronavirus. Penyakit ini mengakibatkan pandemi koronavirus 2019–2020 Penderita COVID-19 dapat mengalami demam, batuk kering, dan kesulitan bernapas. Sakit tenggorokan, pilek, atau bersin-bersin lebih jarang ditemukan. Pada penderita yang paling rentan, penyakit ini dapat berujung pada pneumonia dan kegagalan multiorgan. Infeksi menyebar dari satu orang ke orang lain melalui percikan (droplet) dari saluran pernapasan yang sering dihasilkan saat batuk atau bersin. Waktu dari paparan virus hingga timbulnya gejala klinis berkisar antara 1–14 hari dengan rata-rata 5 hari. (Gorbalenya, Alexander E. 2020).

Kasus positif COVID-19 di Indonesia pertama kali dideteksi pada 2 Maret 2020, ketika dua orang terkonfirmasi tertular dari seorang warga negara Jepang. Pada 9 April, pandemi sudah menyebar ke 34 provinsi dengan Jawa Timur, DKI Jakarta, dan Sulawesi Selatan sebagai provinsi paling terpapar. Sampai tanggal 8 Juli 2020, Indonesia telah melaporkan 68.079 kasus positif, terbanyak di Asia Tenggara melampaui Singapura. Dalam hal angka kematian, Indonesia menempati peringkat keenam terbanyak di Asia dengan 3.359 kematian. Namun, angka kematian diperkirakan jauh lebih tinggi dari data yang dilaporkan lantaran tidak dihitungnya kasus kematian dengan gejala COVID-19 akut yang belum dikonfirmasi atau dites. Sementara itu, diumumkan 31.585 orang telah sembuh, menyisakan 33.135 kasus yang sedang dirawat. (Gorbalenya, Alexander E. 2020).

1. **Black Box**
   1. Pengertian *Black Box*

Menurut budiman (2012), Pengujian black box merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan , dieksekusi pada perangkat lunak diuji apakah telah sesuai dengan yang diharapkan.

Dari kedua defenisi diatas dapat disimpulkan bahwa metode pengujian BlackBox digunakan untuk menguji sistem dari segi user yang dititik beratkan pada pengujian kinerja,spesifikasi dan antarmuka sistem tersebut tanpa menguji kode program yang ada.

* 1. Pengujian BlackBox testing

Menurut Soetam Rizky (2011), Ada beberapa macam metode pengujian Black Box, diantaranya:

1. Equivalence Partioning

*Equivalence Partioning* merupakan metode uji coba Black Box yang membagi domain input dari program menjadi beberapa kelas data dari kasus uji coba yang dihasilkan. Kasus uji penanganan single yang ideal menemukan sejumlah kesalahan (misalnya: kesalahan pemrosesan dari seluruh data karakter) yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati.

1. Boundary Value Analysis

Sejumlah besar kesalahan cenderung terjadi dalam batasan domain input dari pada nilai tengah. Untuk alasan ini boundary value analysis (BVA) dibuat sebagai teknik uji coba. BVA mengarahkan pada pemilihan kasus uji yang melatih nilai-nilai batas. BVA merupakan desain teknik kasus uji yang melengkapi Equivalence partitioning. Dari pada memfokuskan hanya pada kondisi input, BVA juga menghasilkan kasus uji dari domain output.

1. Cause-Effect Graphing Techniques

*Cause-Effect Graphing* merupakan desain teknik kasus uji coba yang menyediakan representasi singkat mengenai kondisi logikal dan aksi yang berhubungan. Tekniknya mengikuti 4 tahapan berikut:

* 1. Causes (kondisi input), dan Effects (aksi) didaftarkan untuk modul dan identifier yang dtujukan untuk masing-masing.
  2. Pembuatan grafik Causes-Effect graph.
  3. Grafik dikonversikan kedalam tabel keputusan.
  4. Aturan tabel keputusan dikonversikan ke dalam kasus uji

1. Comparison Testing

Dalam beberapa situasi (seperti: *aircraft avionic, nuclear Power plant control*) dimana keandalan suatu software amat kritis, beberapa aplikasi sering menggunakan software dan hardware ganda (*redundant*). Ketika software redundant dibuat, tim pengembangan software lainnya membangun versi independent dari aplikasi dengan menggunakan spesifikasi yang sama. Setiap versi dapat diuji dengan data uji yang sama untuk memastikan seluruhnya menyediakan output yang sama. Kemudian seluruh versi dieksekusi secara parallel dengan perbandingan hasil real-time untuk memastikan konsistensi. Dianjurkan bahwa versi independent suatu software untuk aplikasi yang amat kritis harus dibuat, walaupun nantinya hanya satu versi saja yang akan digunakan dalam sistem. Versi independent ini merupakan basis dari teknik Black Box Testing yang disebut Comparison Testing atau back-to-back Testing.

1. **Kerangka Pikir**

Kerangka pikir merupakan hal yang menjadi penopangnya. Sesuai dengan sistem yang berkaitan pada pembahasan ini yaitu sebagai berikut :

Masalah

Diperlukan adanya web service yang dapat memasukkan data perkembangan COVID-19 serta dapat terhubung pada suatu perangkat android.

Data perkembangan COVID-19 pada server yang ada hanya open source pada data Satu Indonesia tidak dalam per-Provinsi.

Kesulitan dalam melakukan pengecekan data yang baru pada setiap saat.

Adanya kebutuhan untuk pengecekan data perkembangan secara detail.

Solusi

Untuk mengantisipasi masalah tersebut. Maka dirancang sebuah web service yang membuat data dapat diolah sebagai sumber informasi serta terhubung dengan suatu perangkat android yang datanya diperbarui setiap saat.

Hasil

Uji coba aplikasi ini dapat membantu masyarakat khususnya pada provinsi Sulawesi Selatan untuk mengetahui lebih update dari perkembangan Covid-19 pada masa Pendemi ini.

# 

# **BAB III**

# **Metode Penelitian**

1. **Waktu dan Tempat Penelitian**

Lama waktu yang akan dibutuhkan dalam penelitian, selama 1 bulan. Untuk memperoleh data-data yang diperlukan penulis akan memilih tempat Dinas Kesehatan yang berada di kota Parepare.

1. **Jenis Penelitian**

Tipe penelitian yang digunakan adalah tipe penelitian deskriptif, yaitu suatu metode dalam meneliti suatu kelompok manusia, suatu objek, suatu kondisi, suatu sistem pemikiran, ataupun suatu peristiwa pada masa sekarang, tujuan dari penelitian deskriptif ini adalah untuk membuat suatu deskripsi, gambaran atau lukisan secara sistematis, faktual, dan akurat mengenai fakta-fakta, sifat-sifat serta hubungan antar fenomena yang diteliti.

1. **Metode Pengumpulan Data**
2. Observasi

Peneliti mengadakan pengamatan secara langsung mengenai sistem yang digunakan dalam menginput member baru dan personal trainer yang digunakan. Setelah semua data terkumpul, baru kemudian data itu di olah lagi.

1. Metode Studi Pustaka

Tahapan pengumpulan data dalam penelitian ini yaitu melalui studi pustaka dengan mengumpulkan data dan informasi dari buku, website dan jurnal yang terkait dengan pokok bahasan penelitian.

1. Wawancara

Wawancara perlu dilakukan untuk mengetahui kebutuhan apa saja yang dibutuhkan untuk menyelesaikan masalah yang telah dipaparkan pada rumusan masalah,juga untuk membuktikan bahwa sistem yang akan dibangun memang dibutuhkan.

1. **Jenis Sumber Data**

Jenis dan sumber data yang dipergunakan dalam penelitian ini adalah sebagai berikut:

1. Data Primer

Data primer adalah data yang diperoleh melalui wawancara maupun pengamatan secara langsung dari sumber data yang bersangkutan dan pengumpulan data-data yang dibutuhkan, cara ini ditempuh agar dapat dipertanggung jawabkan kebenarannya.

1. Data Sekunder

Yaitu data yang diperoleh melalui studi kepustakaan (library research) berupa peraturan Perundangundangan, buku-buku, literature-literatur, laporan hasil penelitian, karya ilmiah, dan sumber lainnya yang berkaitan dengan permasalahan yang diteliti.

1. **Analisis kebutuan**
2. Kebutuhan fungsional

Kebutuhan fungsional merupakan pengertian dari aktivitas yang harus disediakan oleh sistem.yakni:

* + - * 1. Aplikasi menampilkan menu utama yang memiliki beberapa menu pilihan
        2. Aplikasi menampilkan menu pilihan mengenai pendaftaran member
        3. Aplikasi menampilkan menu pilihan mengenai personal trainer

1. Kebutuhan non fungsional

Kebutuhan non fungsional adalah batasan-batasan yang harus disediakan oleh sistem . kebutuhan non fungsional dalam aplikasi ini adalah :

1. Hardware

Dengan spesifikasi sebagai berikut :

1. Prosessor minimal core i5 atau yang setara
2. RAM 4gb atau yang lebih tinggi
3. Harddisk 500gb atau yang lebih tinggi
4. VGA dengan resolusi minimal 800x1200 pixel
5. Software
   * + 1. Android Studio
       2. Operating system minimal windows 8 dan android 4.3 atau yang lebih tinggi
       3. Android versi 7.0 Nougat
       4. XAMPP Versi 5 atau lebih
       5. Visual studio code atau sejenisnya
       6. MySQL 5.5 sebagai database
       7. Navicat untuk pengolahan database
6. Model yang digunakan yaitu UML.
7. **Desain Sistem**
8. Sistem Yang Diusulkan

user

Gambar 3. 1 Sistem Yang Diusulkan

Admin

Sistem yang diusulkan

# 

# **DAFTAR PUSTAKA**

A. Dudhe and S. Sherekar, “Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment,” pp. 1–4, 2014.

A. Rodriguez and I. B. M. Corporation, “RESTful Web services: The basics,” developerWorks, no. February, p. 12, 2008.

B. A. Kasaedja, R. Sengkey, and J. T. Elektro-ft, “Rancang Bangun Web Service Perpustakaan Universitas Sam Ratulangi,” 2014.

Budiman, Agustiar. (2012)."Pengujian Perangkat Lunak dengan Metode Black Box Pada Proses Pra Registrasi User Via Website." Makalah, halaman 4 .

Coronavirus Update [Online] (Live): 307,627 Cases and 13,050 Deaths from\_COVID\_19\_Virus\_Outbreak\_Worldometer". [www.worldometers.info](http://www.worldometers.info).

D. A. R. Anggiani, R. Eko, “Perancangan Sistem Informasi Berbasis Website Subsistem Guru di Sekolah Pesantren Persatuan Islam 99 Rancabango,” Algoritma, vol. 9, pp. 1–11, 2012.

D. Tidwell, “Web Services : The Web â€TM s next Revolution,” IBM Dev. Work., pp. 1–17, 2000.

Dharwiyanti, Sri, and Romi Satria Wahono. (2003). "Pengantar Unified Modeling Language (UML)." Ilmu Komputer . *Environment*. [Online].SanFrancisco.Tersedia:<https://www.getpostman.com/postman>[13 September 2017].

Gorbalenya, Alexander E. (11 Februari 2020). ["Severe acute respiratory syndrome-related coronavirus – The species and its viruses, a statement of the Coronavirus Study Group"](https://www.biorxiv.org/content/10.1101/2020.02.07.937862v1). bioRxiv (dalam bahasa Inggris): 2020.02.07.937862.

JSON. [Online]. Available: http://www.json.org/.

M. Philip and G. Clynch, “A Performance Analysis of WS-\* (SOAP) & RESTful Web Services for implementing Service and Resource Orientated Architectures,” IT&T - 12th Int. Conf. Inf. Technol. Telecommun., no. May, pp. 93–100, 2013.

N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, “Comparison of JSON and XML Data Interchange Formats: A Case Study,” Scenario, vol. 59715, pp. 1–3, 2009.

Noor, E. I. (2016). Sistem Informasi Pendataan Proyek Konstruksi Di PT Bhinneka Karya Cipta.

Postdot Technology. 2017. *Postman is the most complete API Development*

R. Ramanathan and T. Korte, “Software service architecture to access weather data using RESTful web services,” Fifth Int. Conf. Comput. Commun. Netw. Technol., pp. 1–8, 2014.

Rizky, Soetam. (2011)."Pengujian black box." Jakarta: Prestasi Pustaka .

Rothan, H. A.; Byrareddy, S. N. (February 2020). ["The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak"](https://www.researchgate.net/publication/339515532). Journal of Autoimmunity: 102433.

Safaat, Nazruddin. (2012). "Pemrograman aplikasi mobile smartphone dan tablet pc berbasis android." Bandung: informatika .

Setiawan, Eko Budi dan Angga Try R. 2019. “Membangun Aplikasi Android,Web Dan Web Service”. Bandung: Informatika Bandung.

Wulandari, Wulandari, and Siska Aprilia. (2017). "Sistem Informasi Penjualan Produk Berbasis Web Pada Chanel Distro Pringsewu." Jurnal TAM (Technology Acceptance Model) 4 .