**PERENCANAAN PROYEK**

Website Covid-19

Kelompok:

6181801015/Gian Martin D. (Project Manager)

6181801052/Julyus Andreas B. (Kadiv I Frontend )

6181801061/Jason (Kadiv II Backend )

6181801027/Michael Sotaronggal M. (Kadiv III Dokumentasi )

6181801035/Arthur E. Futwembun

6181801044/Yalvi Hidayat

6181801040/David Dimas P.

2016730044/Jessica Natalya

6181801067/Arya Jonathan K.

Tanggal:

11 Desember 2020

1. ****Pendahuluan****
   1. **Tujuan**

Website COVID-19 KorSel ini bertujuan untuk dapa membantu dalam penyampaian informasi seputar COVID-19. Website ini memiliki beberapa halaman yang mencangkup informasi COVID-19 secara umum, data grafik kasus COVID-19, daerah persebaran COVID-19, informasi gejala COVID-19, cara mencegah COVID-19, dan sebagainya.

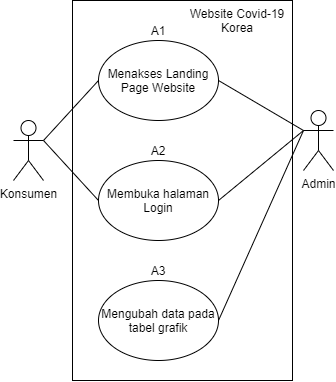
* 1. **Ruang Lingkup**

Sistem yang akan dibangun dapat diakses melalui browser komputer, handphone dan perangkat *mobile* sejenisnya. Sistem dibangun berbasis bahasa pemprogram HTML, PHP dan MySQL, kemudian sistem akan dijalankan di server yang menggunakan sistem operasi Windows 10.

* 1. **Fitur Dasar**

Fitur yang akan ada di dalam sistem ini yaitu,

* Landing page, halaman utama website yang berisi beberapa informasi dasar
* Data grafik, pemaparan data dengan grafik
* Contact Us, halaman pengisian data untuk menghubungi admin
  1. **Use-Case Diagram**

****

* 1. **Use-Case Scenario**

Nomor fungsi: A1

Nama use-case: Mengakses Landing Page Website

Deskripsi: Fitur utama yaitu melihat website

Aktor: Konsumen dan Admin

Pre-kondisi: -

Pos-kondisi: Aktor dapat melihat website sesuai permintaan

Skenario:

1. User membuka browser
2. User menginput url website
3. Sistem mengirim halaman sesuai permintaan user
4. User menerima halaman yang dibuka

Exception:

1. User menginput url website
2. Sistem tidak mengirim halaman sesuai permintaan user
3. Koneksi pada User atau website sedang bermasalah atau keduanya sedang bermasalah

Nomor fungsi: A2

Nama use-case: Mengakses Halaman login

Deskripsi: -

Aktor: Konsumen dan Admin

Pre-kondisi: Aktor sudah ada di halaman index website

Pos-kondisi: Aktor dapat melihat website sesuai permintaan

Skenario:

1. Aktor mengklik tombol login di kanan atas pada bagian header website
2. Sistem mengirimkan halaman login sesuai permintaan aktor
3. Browser menampilkan halaman login

Exception:

1. Aktor sudah login pada website sehingga tidak ada tombol login di kanan atas

Nomor fungsi: A3

Nama use-case: Mengubah data grafik

Deskripsi: -

Aktor: Admin

Pre-kondisi: Aktor sudah melakukan login

Pos-kondisi: Aktor berhasil menambah data pada grafik

Skenario:

1. Aktor membuka halaman admin
2. Sistem mengarahkan aktor pada halaman admin
3. Kemudian sistem memilih tombol data table
4. Aktor menambahakan data pada input box yang tersedia
5. Aktor menekan tombol tambah pada form input
6. Sistem menerima input dan menyimpannya pada MySQL Database

Exception:

1. Aktor membuka halaman admin
2. Sistem mengarahkan aktor pada halaman admin
3. Kemudian sistem memilih tombol data table
4. Aktor menambahakan data pada input box yang tersedia
5. Aktor menekan tombol tambah pada form input
6. Sistem menolak input karena terdapat data yang wajib diisi tetapi masih kosong
7. ****Spesifikasi Teknis****
   1. **Class Diagram HTML**

**A picture containing graphical user interface

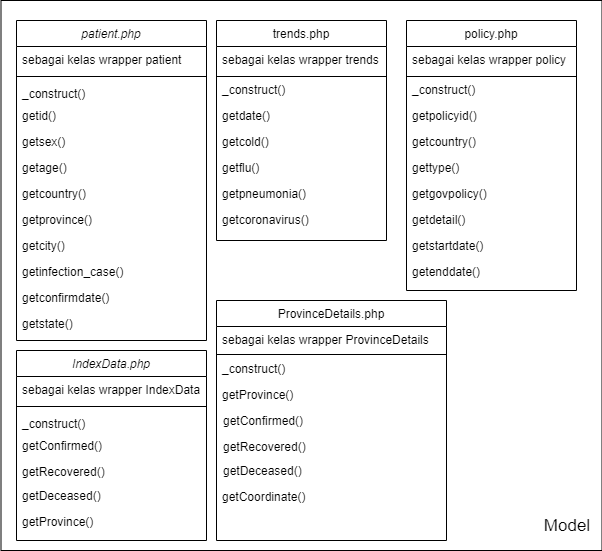
Description automatically generated**

* + 1. **Deskripsi Class**

Pada proyek ini terdapat 1 kelas HTML yaitu pada login.html penjelasannya berikut ini,

**login.html**

* + Head : Atribut ini berisi judul *website* dan juga *link-link* untuk mengimpor css *stylesheet*-nya.
  + Body : Atribut ini berisi konten dalam bentuk tipe data String dan SVG.
  + Submit : Method yang berfungsi mengirimkan input data user dengan parameter bertipe data String.
  1. **Class Diagram PHP**

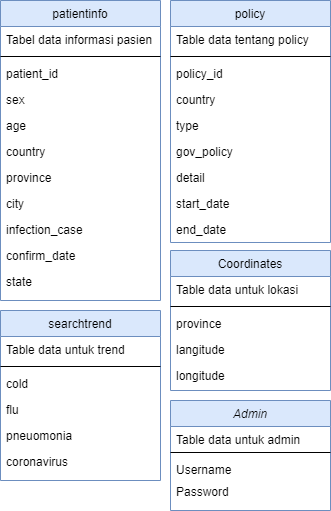




**Graphical user interface, table

Description automatically generated**

* + 1. **Deskripsi Class**
* **Model**
* **patient.php** : berfungsi sebagai kelas wrapper *patient* yang menampilkan data pasien (patient\_id, sex, age, country, province, city, infection\_case, confirmed\_date, state)
* **trend.php** : berfungsi sebagai kelas wrapper *trends* yang menampilkan data trend (date, cold, flu, pneumonia, coronavirus)
* **policy.php** : berfungsi sebagai kelas wrapper *policy* yang menampilkan data policy (policy\_id, country, type, gov\_policy, detail, start\_date, end\_date)
* **ProvinceDetails.php** : berfungsi sebagai kelas wrapper *ProvinceDetails* yang menampilkan data detail province (province, confirmed, recovered, deceased, coordinate)
* **IndexData.php** : berfungsi sebagai kelas wrapper *IndexData* yang menampilkan data index (province, confirmed, recovered, deceased)
* **View**
* **about.php** : berfungsi sebagai kelas wrapper *about* yang menampilkan informasi umum COVID-19 (definisi, gejala, pencegahan, pengobatan)
* **admin.php** : berfungsi untuk menampilkan halaman data untuk admin
* **data.php** : berfungsi sebagai kelas wrapper *data* yang menampilkan data COVID-19 dalam chart, tabel, dan lokasi
* **datacharts.php** : berfungsi untuk menampilkan data charts
* **datatable.php** : berfungsi untuk menampilkan data tabel
* **login.php** : berfungsi untuk menampilkan halaman login admin
* **layout.php** : berfungsi sebagai pengaturan layout website
* **index.php** : berfungsi sebagai kelas wrapper *home*
* **Controller**
* **adminController.php** : berfungsi untuk memberi admin kontrol atas halaman admin
* **dataController.php** : berfungsi untuk mengontrol (menambah/mengurangi/mengubah) data pada tabel maupun charts
* **indexController.php** : berfungsi sebagai navigasi pada website
* **viewService.php** : berfungsi sebagai kelar wrapper pada index data
* **mysqlDBService.php** : berfungsi sebagai penghubung antara php dan database
  + 1. **Deskripsi Method**
* **Model**
* **IndexData.php :**
* \_construct() : untuk memodelkan kelas IndexData
* getConfirmed() : getter dari variabel confirmed pada Indexdata.php
* getRecovered() : getter dari variabel recovered pada Indexdata.php
* getDeceased() : getter dari variabel deceased pada Indexdata.php
* getProvince() : getter dari variabel province pada Indexdata.php
* **patient.php :**
* \_construct() : untuk memodelkan kelas patient.php
* getid() : getter dari variabel id pada patient.php
* getsex() : getter dari variabel sex pada patient.php
* getage() : getter dari variabel age pada patient.php
* getcountry() : getter dari variabel country pada patient.php
* getprovince() : getter dari variabel province pada patient.php
* getcity() : getter dari variabel city pada patient.php
* getinfection\_case() : getter dari variabel infection\_case pada patient.php
* getconfirmdate() : getter dari variabel confirmdate pada patient.php
* getstate() : getter dari variabel state pada patient.php
* **policy.php :**
* \_construct() : untuk memodelkan kelas policy.php
* getpolicyid() : getter dari variabel policyid pada policy.php
* getcountry() : getter dari variabel country pada policy.php
* gettype() : getter dari variabel type pada policy.php
* getgovpolicy() : getter dari variabel govpolicy pada policy.php
* getdetail() : getter dari variabel detail pada policy.php
* getstartdate() : getter dari variabel startdate pada policy.php
* getenddate() : getter dari variabel enddate pada policy.php
* **ProvinceDetails.php :**
* \_construct() : untuk memodelkan kelas ProvinceDetails.php
* getProvince() : getter dari variabel Province pada ProvinceDetails.php
* getConfirmed() : getter dari variabel Confirmed pada ProvinceDetails.php
* getRecovered() : getter dari variabel Recovered pada ProvinceDetails.php
* getDeceased() : getter dari variabel Deceased pada ProvinceDetails.php
* getCoordinate() : getter dari variabel Coordinate pada ProvinceDetails.php
* **trends.php :**
* \_construct() : untuk memodelkan kelas trends.php
* getdate() : getter dari variabel date pada trends.php
* getcold() : getter dari variabel cold pada trends.php
* getflu() : getter dari variabel flu pada trends.php
* getpneumonia() : getter dari variabel pneumonia pada trends.php
* getcoronavirus() : getter dari variabel coronavirus pada trends.php
* **Controller**
* **adminController.php :**
* ViewLogin() : berfungsi untuk membuat halaman login terhubung dengan viewService.php
* login() : berfungsi untuk masuk/login dari user admin pada login page
* getAllPatient() : untuk menampilkan data pasien dari database yang dimodelkan dengan patient.php
* getAllTrend() : untuk menampilkan data trend dari database yang dimodelkan dengan trend.php
* getAllPolicy : untuk menampilkan data policy dari database yang dimodelkan dengan policy.php
* view\_Admin() : berfungsi untuk membuat halaman admin terhubung dengan viewService.php
* logout() : berfungsi untuk keluar/logout dari user admin
* add\_patient() : berfungsi untuk menambah pasien baru
* getid() : berfungsi untuk mengambil id terakhir dari database tabel patient yang akan digunakan di method add\_patient()
* getidp() : berfungsi untuk mengambil id terakhir dari database tabel policy yang akan digunakan di method add\_policy()
* add\_searchtrend() : berfungsi untuk menambah trend baru ke database
* add\_policy() : berfungsi untuk menambah policy baru ke database
* \_contstruct() : untuk memanggil kelas MySQLDB untuk membuka koneksi ke database covid19korsel
* **dataController.php :**
* view\_data() : berfungsi untuk menampilkan halaman data.php
* view\_table() : berfungsi untuk menampilkan halaman datatables.php
* view\_charts() : berfungsi untuk menampilkan halaman datacharts.php
* patientinfo() : untuk mengambil semua data dari tabel patientinfo
* searchtrends() : untuk mengambil semua data dari tabel searchtrend
* govpolicy() : untuk mengambil semua data dari tabel policy
* chart1() : berfungsi untuk menampilkan tahun dari data kategori “trend” beserta jumlah rata-rata dari tiap tahun beserta “trend” dari tabel searchtrend
* chart2() : berfungsi untuk menampilkan tanggal dari data kategori “trend” beserta jumlah rata-rata kategori coronavirus dari tabel searchtrend
* chart3() : berfungsi untuk menampilkan data kategori “age” beserta jumlah dari masing-masing “sex” dari tabel patientinfo
* chart4() : berfungsi untuk menampilkan data kategori “sex” beserta jumlah dari masing-masing “sex” dari tabel patientinfo
* chart5() : berfungsi untuk menampilkan data kategori “state” beserta jumlah dari masing-masing “state” dari tabel patientinfo
* \_contstruct() : untuk memanggil kelas MySQLDB untuk membuka koneksi ke database covid19korsel
* **indexController.php :**
* view\_index() : berfungsi untuk menampilkan halaman index.php
* view\_about() : berfungsi untuk menampilkan halaman about.php
* getIndexData() : berfungsi untuk menampilkan jumlah dari data tabel “state : released; deceased” dan “province” dari database
* getProvinceDetails() : berfungsi untuk menampilkan daftar province beserta total dari released, deceased, beserta koordinat langitude dan longitude dari tabel patientinfo
* \_contstruct() : untuk memanggil kelas MySQLDB untuk membuka koneksi ke database covid19korsel
* **viewService.php :**
* createView() : untuk membuat view terhubung dengan layout.php dan menampilkan parameter bawaan dari controller
* createView2() : untuk membuat view terhubung dengan layout.php
* createView3() : untuk menampilkan view tanpa layout dan parameter bawaan pada controller
* createView4() : untuk menampilkan view
* **mysqlDBService.php :**
* openConnection() : untuk membuka koneksi pada database yang akan digunakan pada method \_construct()
* executeSelectQuery() : berfungsi untuk mengambil data pada tabel berdasarkan kueri yang dipilih
* executeNonSelectQuery() : berfungsi untuk menambah/mengupdate data pada tabel
* getConnection() : untuk mendapatkan koneksi ke databse
* escapeString() : berfungsi untuk menghilangkan karakter khusus dalam data String agar dapat digunakan dalam statement kueri SQL
* lostConnection() : untuk menutup koneksi pada database yang akan digunakan pada method executeSelectQuery() dan executeNonSelectQuery() pada mysqlDBService.php
* ngitungbaris() : berfungsi untuk menghitung baris dari kueri yang dipilih
* betulga() : berfungsi untuk mengecek apakah suatu kondisi benar atau tidak pada method yang akan digunakan pada method login() pada adminController.php
* \_contstruct() : untuk memodelkan kelas MySQLDB
  1. **Class Diagram Tabel MySQL**

****

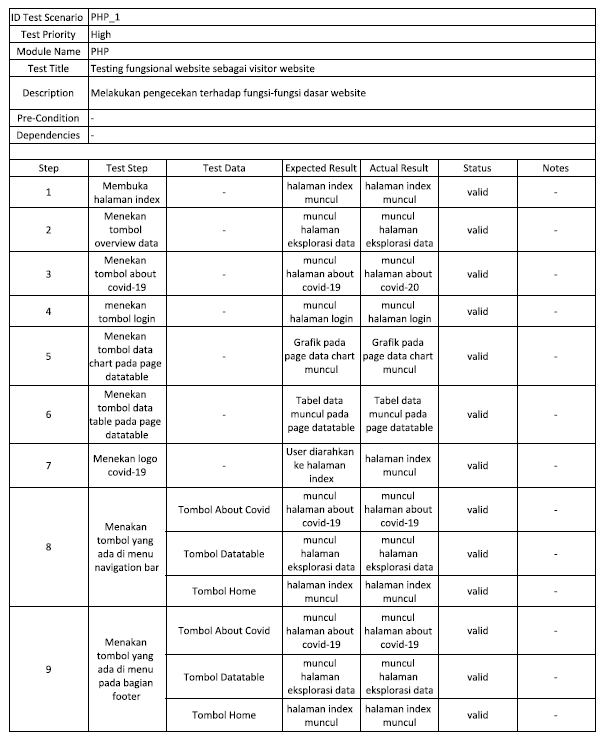
* + 1. **Deskripsi Database**
* **patientifo :** menyimpan tabel data yang berisikan informasi pasien (patient\_id, sex, age, country, province, city, infection\_case, confirm\_date, state)
* **policy :** menyimpan tabel data tentang policy (policy\_id, country, type, gov\_policy, detail, start\_date, end\_date)
* **searchtrend :** menyimpan tabel data untuk trend (cold, flu, pneumonie, coronavirus)
* **Coordinates :** menyimpan tabel data untuk lokasi (province, langitude, longitude)
* **Admin :** menyimpan tabel data informasi login admin (Username, Password)

1. ****Dokumentasi Testing****
   1. **HTML Test Case**

**Table

Description automatically generated**

* 1. **PHP Test Case**

****

* 1. **MySQL Test Case**

**Table

Description automatically generated**

1. ****Kesimpulan****

Fitur dasar dan spesifikasi yang disepakati diawal sudah diselesaikan sesuai dengan waktu yang ditentukan. Kemudian semua pekerjaan sudah di dokumentasikan melalui Use Case Diagram, Class Diagram dan Test Case. Pada Bab 1 terdapat semua pengenalan mengenai website ini mulai dari daftar fitur dasar hingga skenario penggunaannya.

Pada Bab 2 dijelaskan mengenai modul-modul yang ada pada website ini melalui class diagram dan penjelasan detail method-methodnya. Setelah proyek diselesaikan pada bab 3 dilakukan testing pada jenis modul yang dibagi 3 yakni, HTML, PHP, dan MySQL. Pada Bab 3 terdapat dokumen test case dengan beberapa eksperimen beserta hasil dari eksperimen tersebut (sesuai/tidak).

Melalui hasil dari test case yang telah kami lakukan, dapat kami simpulkan bahwa website ini sudah siap digunakan oleh klien, dan website ini masih mempunyai potensi untuk dikembangkan lebih lanjut dikemudian hari.

1. ****Referensi****

Use Case Diagram:

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

Class Diagram:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

Testing Documentation:

https://www.softwaretestinghelp.com/test-case-template-examples/