

RANCANG BANGUN APLIKASI BIOINFORMATIKA BOKIBA MENGUNAKAN BIOPYTHON DAN FRAMEWORK LAMINAS

Edrian Hadinata, S.Kom, M.Kom

Universitas Harapan Medan

Email: edrianhadinata@gmail.com

ABSTRACT

Currently, there are various emerging applications of bioinformatics that can make it easier for biological researchers to be able to analyze DNA and RNA sequences. Biological applications such as Biopython are open source module-based applications that can be combined with other programs for free. Many web-based applications have also been developed because they are considered easier than desktop applications because they are portable in addition to being easier to integrate with central bioinformatics data around the world. Laminas Framework which carries the MVC concept and is based on object-oriented programming and with its advantages fits the pattern of bioinformatics application development. The hybrid process between the system architecture contained in the framework and business processes in bioinformatics is a challenge for researchers. The process of data communication and execution of shell commands that are triggered in the middleware and continuously connected to the data source makes the process faster. The results of testing the laminas framework through the phpunit test contained in the laminas framework can be tested well and the python module that is formed is tested with pytest and unittest displays good results. so that structured and flexible bioinformatics applications can be developed further.

Keywords: Bioinformatics, Laminas Framework, Biopython

PENDAHULUAN

Penemuan urutan DNA yang lebih cepat terjadi pada pertengahan tahun 1970 an saat itulah terjadi ledakan jumlah urutan DNA yang dapat diungkapkan tahun 1980 an dan 1990 an. Hal ini menjadi penyebab munculnya proyek-proyek pengungkapan genom yang kemudian menjadi landasan betapa perlunya pengelolaan data analisis urutan yang pada akhirnya menjadi penyebab lahirnya bidang ilmu baru Bioinformatika(Witarto, 2003).

Bioinformatika sendiri diciptakan oleh Paulien Hogeweg and Ben Hesper tahun 1970 (Abdurakhmonov, 2016). Aplikasi Bioinformatika yang berkaitan erat dengan penggunaan pangkalan data urutan biologi adalah BLAST(Basic Local Alignment Search Tool)(Gupta, 2010). Melalui penggunaan BLAST, ilmuwan memungkinkan melakukan pencarian urutan baik asam nukleat maupun protein yang mirip dengan urutan tertentu.(Pratiwi, 2018)

Adapun aplikasi Bioinformatika lain yang digunakan untuk proses pembacaan genome virus, misalnya seperti adalah virus Corona antara lain adalah Webprimer yang disediakan oleh Stanford Genomic Resources, GenWalker yang disediakan oleh

Cybergene AB (Nugroho et al, 2018). Aplikasi tersebut pun dapat digunakan secara gratis, meskipun demikian juga terdapat aplikasi yang komersial seperti Primer Designer yang di kembangkan oleh Scientific & Education Software, dan software-software untuk analisa DNA lainnya seperti Squencher, SeqMan II, Genetyx, DNASIS dan lain-lain.

Aplikasi biologi seperti UGENE, EMBOSS, GenGIS, GENTle, MOTHUR, BioPerl, PathVisio, BioJava, GenoCAD, Biopython, GeWorkbench, GenomeSpace, Bioclipse, .NET Bio, Apache Taverna, BioJS, Bioconductor, and BioRuby (Goto et al, 2010) merupakan aplikasi berbasis modul modul open source yang secara gratis dan dapat dikombinasikan program yang lain (Abdurakhmonov, 2016).

Aplikasi Biopython sendiri dirancang untuk menghasilkan objek python melalui proses parsing dari beragam algoritma penjajaran atau pengelompokan, dan untuk berbagai tugas hilir termasuk klasifikasi. BioPython juga menyediakan infrastruktur untuk dekomposisi tugas yang dapat diparalelkan menjadi proses yang dapat dipisahkan untuk komputasi pada sekelompok stasiun kerja (Gentleman, 2004). Terkait dengan pembuatan aplikasi Bioinformatika itu sendiri, jika aplikasi tersebut berjalan di protokol http atau https tentu saja aplikasi tersebut berjalan selayaknya website (Gomez-Casati, 2017), menggunakan web server, menjalankan program server side dan servis lainnya serta servis yang berhubungan dengan pemanfaatan database (Stephenson et al, 2018). Begitupun dalam pembuatan aplikasi bioinformatika berbasis web tidaklah selalu dirancang dari kondisi nol atau dengan kata lain dimana proses pengembangan dimulai dengan menggunakan program yang telah tersusun dan memiliki kerangka konsep yang terstruktur rapi, seperti Web Framework.

Zend Framework atau sekarang disebut Laminas, Laminas memiliki keunggulan dibandingkan dengan framework lain, yaitu Laminas dibuat oleh pengembang yang mengembangkan engine PHP, memiliki dokumentasi yang lengkap, library yang lengkap untuk memenuhi kebutuhan pengembang aplikasi, mendukung pengembangan MVC(Model View Controller) (Supaartagorn, 2011) (Maulana et al, 2019). Selain itu, struktur dan komponen Laminas dirancang dengan beberapa dependensi pada komponen lain, fleksibel dan keunggulan yang lain adalah arsitektur dapat ditambah sehingga pengembang memungkinkan menggunakan komponen secara individual. Kekurangannya ialah lebih rumit dalam proses penggunaan disebabkan oleh dependensi begitupun proses tersebut terkait dengan sistem keamanan yang terdapat pada Laminas, yang lain adalah keterbatasan pengetahuan developer terhadap bidang ilmu bioinformatika itu sendiri begitupun kekurangan ini tertutupi oleh ketersediaan informasi yang terarsip dengan baik melalui paper bioinformatika(Lawlor, 2015).

Oleh karena itu Laminas Framework dengan keunggulannya cocok dengan pola pengembangan aplikasi bioinformatika seperti Biopython yang memiliki pola pengembangan pencarian, analisis, prediksi serta clusterisasi dari domain ilmu biologi yang mungkin dapat dikembangkan dengan pola arsitektur sendiri. Penelitian ini membahas tentang bagaimana menghybrid dua konsep arsitektur menjadi satu kesatuan. Laminas dengan MVC dengan konsep dependensi sedangkan aplikasi bioinformatika dengan pola analisis, prediksi dan klasterisasi.

LANDASAN TEORI

Laminas Framework

Laminas adalah sub proyek dari Linux Foundation yang merupakan transisi dari Zend Framework yang diumumkan April 2019 namun secara resmi dimulai sejak Januari 2020. Seperti juga Zend Framework (Romer, 2016). Laminas berisikan modul program profesional dari bahasa pemrograman PHP yang telah diinstall di seluruh dunia lebih dari 570 juta instalasi. Modul lamina digunakan untuk mengembangkan aplikasi dan layanan berbasis web dan menggunakan 100% kode pemrograman berbasis objek (Framework.zend.com, 2021).

Biopython

Biopython menggunakan bahasa pemrograman python yang telah secara luas dikenal oleh banyak pengembang sebagai bahasa pemrograman tingkat tinggi. Python dan Biopython bebas di download secara gratis (Hamelryck et al, 2003). Proyek Biopython adalah proyek kolaborasi yang telah lama yang dilakukan oleh para relawan pengembang aplikasi (Chang, 2020), menyediakan pustaka bahasa pemrograman python untuk berbagai permasalahan bioinformatika (Moitra, 2015). Biopython berisi modul untuk membaca dan menulis file penjajaran yang berbeda format dan beragam penjajaran urutan, dapat mengerjakan struktur 3D macromolecular, dapat berinteraksi dengan perangkat aplikasi seperti BLAST, ClustalW dan EMBOSS, pengaksesan kunci database secara online, baik juga dalam penyediaan sejumlah metode pembelajaran secara statistik. (Cock PJ, 2009).

Doctrine ORM 2(Object Relationship Mapping)

Object relationship mapping adalah sebuah teknologi yang menjadi satu lapis antara aplikasi dengan database yang mana ORM bekerja seperti database, tetapi hanya berbentuk objek. Setiap objek pada umumnya mewakili tabel didalam database (Lorenz, 2016).

Pytest

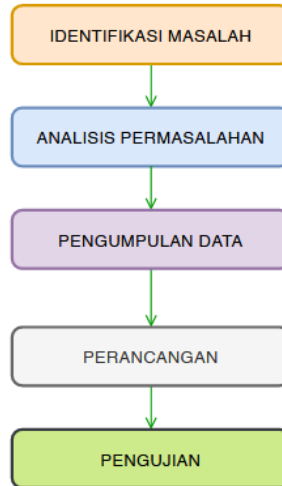
Pytest adalah alat bantu atau kerangka kerja pengujian pada kode python yang memudahkan untuk melakukan pengujian sederhana (Pajankar, 2017).

National Center for Biotechnology Information(NCBI)

NCBI adalah server bank data berisikan database yang memuat ribuan dataset berkaitan dengan informasi kesehatan dan bioteknologi (Benson, 2018). Dataset terus menerus di update sesuai dengan hasil riset terbaru yang berisi DNA, Protein, Senyawa aktif dan taksonomi. Disamping penyediaan dataset, NCBI juga berisi berbagai macam aplikasi untuk analisis DNA, protein 3D, pencarian primer, pencarian conserve domain dan lain sebagainya. NCBI merupakan salah satu bankdata gen, protein dan literaturnya (Federhen, 2012).

METODE

Proses penyelesaian masalah dalam penelitian ini diuraikan dalam alur kerja penelitian antara lain:



Gambar 1. Kerangka alur kerja

Identifikasi Masalah

Dalam proses ini penulis mengumpulkan beragam permasalahan terkait dengan pengimplementasian framework laminas dan biopython.

Analisis Permasalahan

Dalam proses ini penulis mengumpulkan beragam permasalahan terkait dengan pengimplementasian framework laminas dan biopython.

Rancangan Pengumpulan Data

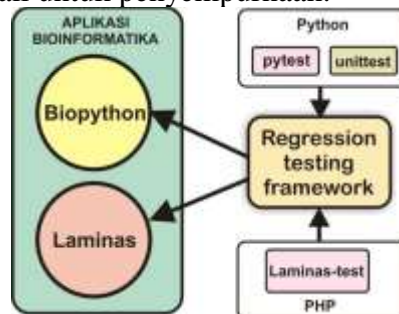
Adapun pengumpulan data pada penelitian ini antara lain, sebagai berikut:

Tabel 3. Pengumpulan Data

No	Tinjauan	Sumber
1	Dataset Bioinformatika	https://www.ncbi.nlm.nih.gov/
2	Biopython	https://github.com/biopython/biopython
3	Framework Laminas	https://github.com/laminas/laminas-mvc
4	Python	https://www.python.org/downloads/

Rancangan Pengujian

Pada tahap ini dilakukan pengujian terhadap aplikasi yang telah dihasilkan dan pengkajian kembali kelayakan aplikasi tersebut sudah sesuai atau masih perlu dilakukan peninjauan kembali untuk penyempurnaan.



Gambar 6. Rancangan Pengujian

HASIL DAN PEMBAHASAN

Identifikasi Masalah

Identifikasi masalah pada penelitian ini dirangkum dan dapat dilihat pada tabel dibawah ini:

Tabel 1. Identifikasi Masalah

No	Masalah
1	Arsitektur sistem pada Framework Laminas mengguna konsep Model View Controller dimana Model berisikan modul-modul program yang di bentuk menggunakan konsep pemrograman berorientasi objek yang memetakan kondisi tabel yang terdapat pada database dan struktur data dari hasil bisnis logic yang terdapat pada sistem. Sementara data yang diproses pada Aplikasi Bioinformatika merupakan raw data yang berasal dari dataset biologi para ilmuwan yang berisi data DNA, RNA dll.
2	Bisnis logic yang terdapat pada Aplikasi Bioinformatika berisikan alur penelitian biologi yang terkadang berurutan untuk satu jenis proses atau memiliki tahapan dan berpindah dari proses satu ke proses lainnya hingga menghasilkan satu output analisis atau prediksi. Sementara Model Arsitektur Framework Laminas hanya memiliki standar MVC, begitupun akan susah jika tiap jenis proses dikumpulkan menjadi satu modul.
3	Integrasi MVC pada Laminas Framework terhadap bahasa pemrograman python yang tidak disupport oleh modul-modul PHP dan Web Server menjadikan keduanya berjalan terpisah

Analisis Permasalahan

Dengan menganalisa masalah yang telah ditentukan sebelumnya, maka diharapkan masalah dapat dipahami dengan baik.

Adapun Analisis permasalahan yang diperoleh terlihat pada tabel dibawah ini:

Tabel 2. Analisis Permasalahan

No	Masalah	Solusi
1	Arsitektur sistem pada Framework Laminas mengguna konsep Model View Controller dimana Model berisikan modul-modul program yang di bentuk menggunakan konsep pemrograman berorientasi objek yang memetakan kondisi tabel yang terdapat pada database dan struktur data dari hasil bisnis logic yang terdapat pada sistem. Sementara data yang diproses pada Aplikasi Bioinformatika merupakan raw data yang berasal dari dataset biologi para ilmuwan yang berisi data DNA, RNA dll.	Model yang berisikan data komponen dari dataset biologi dimasukkan kedalam database tetapi hanya meta datanya saja sementara File Raw Dataset diupload dan disimpan ke dalam server. Content data yang diambil dari raw file dikonversi kedalam bentuk JSON untuk bisa digunakan secara lebih luas.

2	Bisnis logic yang terdapat pada Aplikasi Bioinformatika berisikan alur penelitian biologi yang biasanya berurutan untuk satu jenis proses atau memiliki tahapan dan berpindah dari proses satu ke proses lainnya hingga menghasilkan satu output analisis atau prediksi. Sementara Model Arsitektur Framework Laminas hanya memiliki standar MVC, begitupun akan susah jika tiap jenis proses dikumpulkan menjadi satu modul.	Buat modul untuk setiap jenis proses, namakan modul tersebut dengan nama prosesnya, modul tersebut memiliki Model, Controller dan View Sendiri.
3	Integrasi MVC pada Laminas Framework terhadap bahasa pemrograman python yang tidak disupport oleh modul-modul PHP dan Web Server menjadikan keduanya berjalan terpisah	Instal Modul Python didalam Server dan instal juga Biopython. Setelah output testing dinyatakan berhasil. Maka buat modul Logic sehingga model MVC berubah menjadi MVLC (Model View Logic Controller). Modul logic berisikan class untuk menjalankan perintah python melalui perintah shell exec.

Pengumpulan Data

Dataset

Data yang digunakan pada aplikasi adalah data genome yang diperoleh dari situs penerima hibah data dari para peneliti salah satunya NLM (National Library of Medicine). Khusus untuk data yang dipergunakan pada rancangan ini, adalah data yang dipergunakan bidang ilmu bioinformatika. Untuk isi file terlihat pada tampilan berikut ini:

```
>lc1|MT772403.1_cds_QLI49919.1_1 [gene=ORFlab] [protein=ORFlab polyprotein]
[exception=ribosomal slippage] [protein_id=QLI49919.1]
[location=join(212..13414,13414..21501)] [gbkey=CDS]
ATGGAGAGCCTTGTCCCTGGTTTCAACGAGAAAACACACGTCCAACCTCAGTTTGCCTGTTTTACAGGTTTC
GCGACGTGCTCGTACGTGGCTTTGGAGACTCCGTGGAGGAGTCTTATCAGAGGCACGTCAACATCTTAA
AGATGGCACTTGTGGCTTAGTAGAAGTTGAAAAAGGCGTTTTGCCTCAACTGAAACAGCCCTATGTGTTTC
ATCAAACGTTTCGGATGCTCGAACTGCACCTCATGGTCATGTTATGGTTGAGCTGGTAGCAGAACTCGAAG
GCATTTCAGTACGGTTCGTAGTGGTGAGACACTTGGTGTCTTGTCCCTCATGTGGGCGAAATACCAGTGGC
```

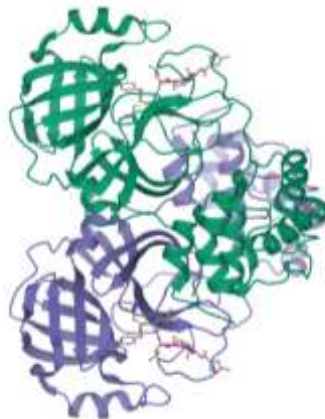
Ini adalah data Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1. Data tersebut adalah sampel data mentah genome yang diperoleh dari website ncbi.nlm.gov.id dalam format txt. Tentunya tidak semua bagian diambil untuk analysis. Ada bagian-bagian tertentu yang akan diekstraksi. Seperti data struktur sekuensial DNA Adenine(A), Guanine(G), Thymine(T) dan Cytosine(C). Selain itu data yang diupload untuk kepentingan aplikasi bioinformatika adalah data berekstensi .pdb yang berisi struktur kristal enzim protease yang nantinya digunakan untuk membentuk grafik. Isi file tersebut adalah sebagai berikut.

```

HEADER VIRAL PROTEIN 26-JAN-20 6LU7
TITLE THE CRYSTAL STRUCTURE OF COVID-19 MAIN PROTEASE IN COMPLEX WITH AN
TITLE 2 INHIBITOR N3
COMPND MOL_ID: 1;
COMPND 2 MOLECULE: MAIN PROTEASE;
COMPND 3 CHAIN: A;
COMPND 4 FRAGMENT: 3C-LIKE PROTEINASE;
COMPND 5 SYNONYM: PP1AB,ORF1AB POLYPROTEIN;
COMPND 6 EC: 3.4.22.69;
COMPND 7 ENGINEERED: YES;
COMPND 8 MOL_ID: 2;
COMPND 9 MOLECULE: N-[(5-METHYLISOXAZOL-3-YL)CARBONYL]ALANYL-L-VALYL-N~1~-
COMPND 10 ((1R,2Z)-4-(BENZYLOXY)-4-OXO-1-[[ (3R)-2-OXOPYRROLIDIN-3-
COMPND 11 YL]METHYL}BUT-2-ENYL)-L-LEUCINAMIDE;
COMPND 12 CHAIN: C;
COMPND 13 ENGINEERED: YES
SOURCE MOL_ID: 1;
SOURCE 2 ORGANISM_SCIENTIFIC: SEVERE ACUTE RESPIRATORY SYNDROME
CORONAVIRUS
SOURCE 3 2;
SOURCE 4 ORGANISM_COMMON: 2019-NCOV;
SOURCE 5 ORGANISM_TAXID: 2697049;
SOURCE 6 GENE: REP, 1A-1B;
SOURCE 7 EXPRESSION_SYSTEM: ESCHERICHIA COLI BL21(DE3);
SOURCE 8 EXPRESSION_SYSTEM_TAXID: 469008;
SOURCE 9 EXPRESSION_SYSTEM_VECTOR_TYPE: PLASMID;
SOURCE 10 EXPRESSION_SYSTEM_PLASMID: PGEX-6P-1;
SOURCE 11 MOL_ID: 2;
SOURCE 12 SYNTHETIC: YES;

```

Kedua file tersebut diupload secara hierarchy. File pertama yang diupload menjadi main file yang akan dianalisis dan file kedua adalah file tambahan yang digunakan untuk kepentingan lain dalam rangka mengambil informasi tambahan. Contohnya. File pdb tersebut dan data struktur DNA akan berguna untuk menghasilkan gambar berikut.



Gambar 2. Biological Assembly

Pada proses pengambilan data pada aplikasi bioinformatika dengan menggabungkan framework ini dilakukan dengan dua cara seperti yang telah disebutkan sebelumnya, yaitu:

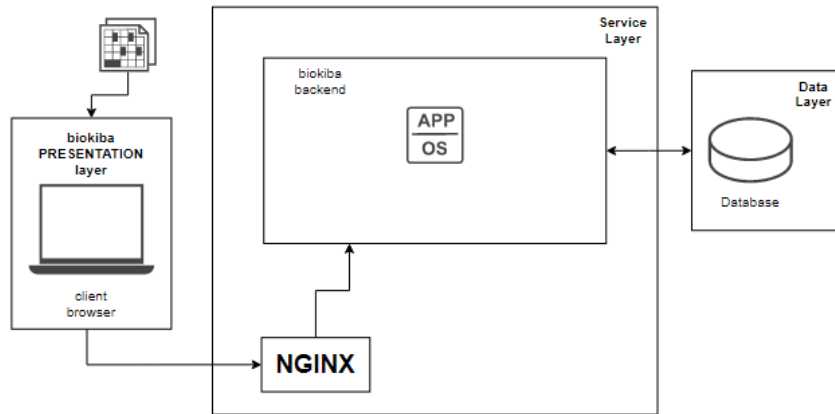
1. Sistem Upload File

2. Memanfaatkan API (Application Programming Interface)

Dua cara pemanfaatan ini dilakukan dengan memanfaatkan framework Laminas. Pemanfaatan secara upload dilaksanakan jika file diupload secara terpisah.

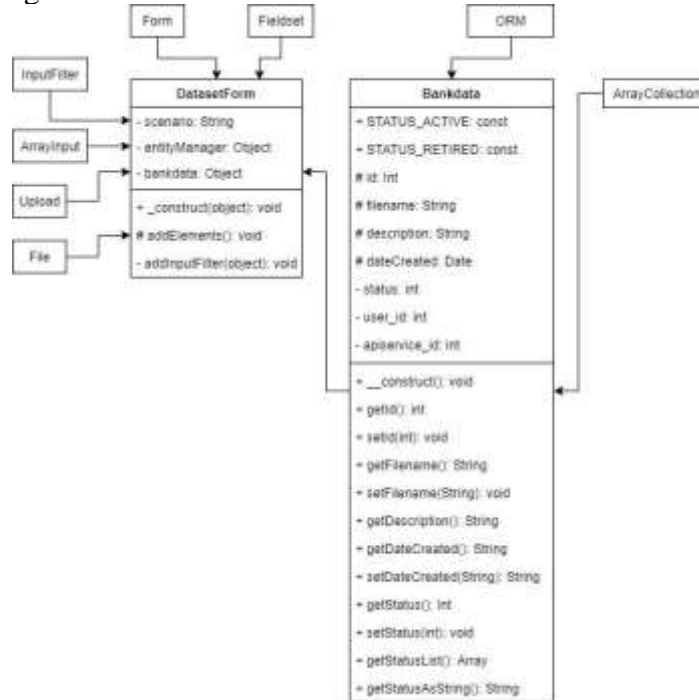
Perancangan Pengumpulan Data Manual dan API

Perancangan sistem unggah dataset atau upload secara manual dilaksanakan melalui form upload. File yang dapat diupload berextensi .txt, .csv, .fasta dan .pdb. Untuk tampilan rancangan arsitekturnya antara lain sebagai berikut:



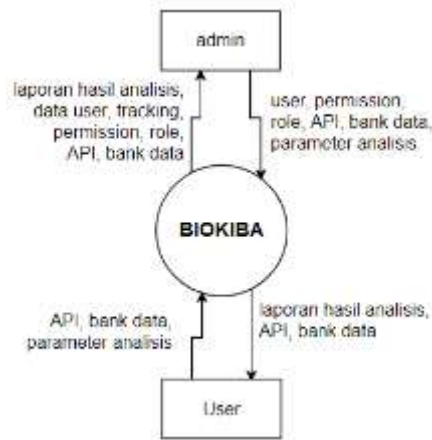
Gambar 3. Perancangan unggah dataset

Untuk rancangan dan implementasinya pada framework Laminas terlihat pada gambar class diagram Bankdata dan DatasetForm berikut ini:



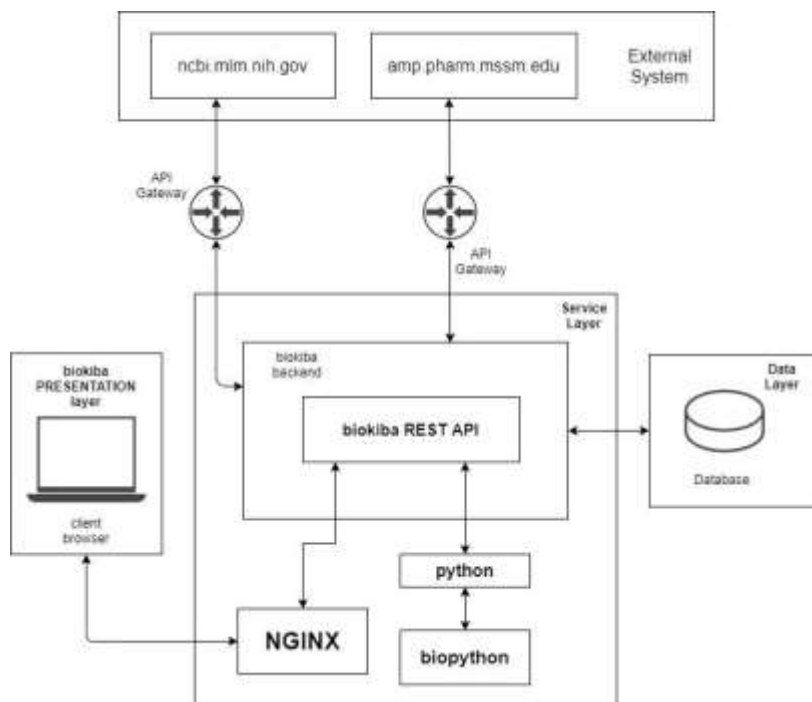
Gambar 4. Class diagram bankdata

Sebagai gambaran arus informasi yang masuk dan yang diberikan ke admin dan server oleh sistem ditampilkan pada data flow diagram berikut ini.



Gambar 5. Diagram Konteks

Untuk perancangan pemanfaatan data melalui data sering menggunakan situs penyedia data dengan memanfaatkan API(Application Programming Interface). Pemanfaatan dengan pemanggilan data melalui URL dengan memanfaatkan web service yang berasal dari external juga dilakukan dalam rangka pemanggilan data. Untuk diagram arsitekturnya terlihat seperti pada gambar dibawah ini.



Gambar 6. Arsitektur Pengambilan dataset

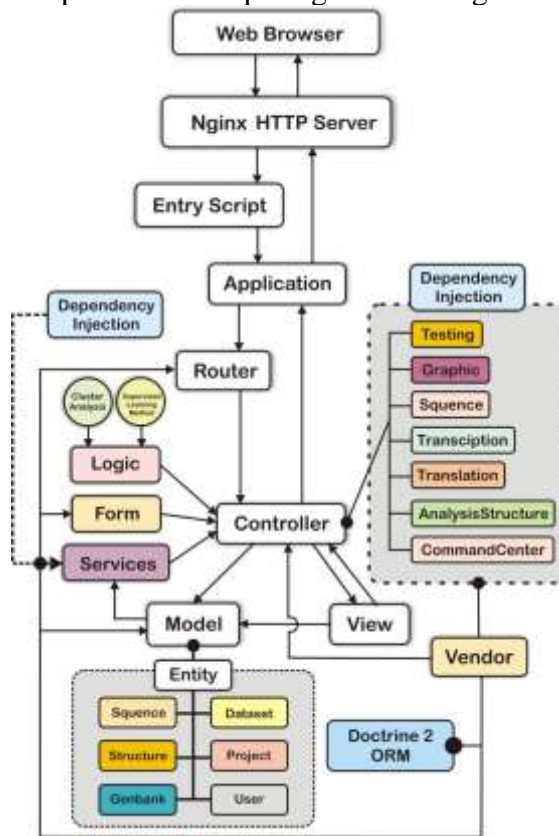
Pemanfaatan API dalam rangka memberikan pelayanan pertukaran data sangat penting dalam mengintegrasikan dan mengizinkan dua aplikasi atau lebih yang berbeda bisa saling terhubung bersamaan.

API disediakan oleh platform aplikasi di situs <http://ncbi.nlm.nih.gov>. Aplikasi National Center for Biotechnology Information(NCBI) tersebut mengizinkan pengguna mengakses data melalui platform web service.

Proses pemanfaatan aplikasi tambahan untuk memudahkan pengembangan aplikasi pun dilakukan diantaranya dengan menginstall eutils 0.6.0 didalam server. Aplikasi ini bermanfaat sebagai interface dalam paket pemrograman python untuk memudahkan proses pencarian, pengambilan dan proses parsing baris data didalam dataset yang tersedia di situs ncbi tersebut

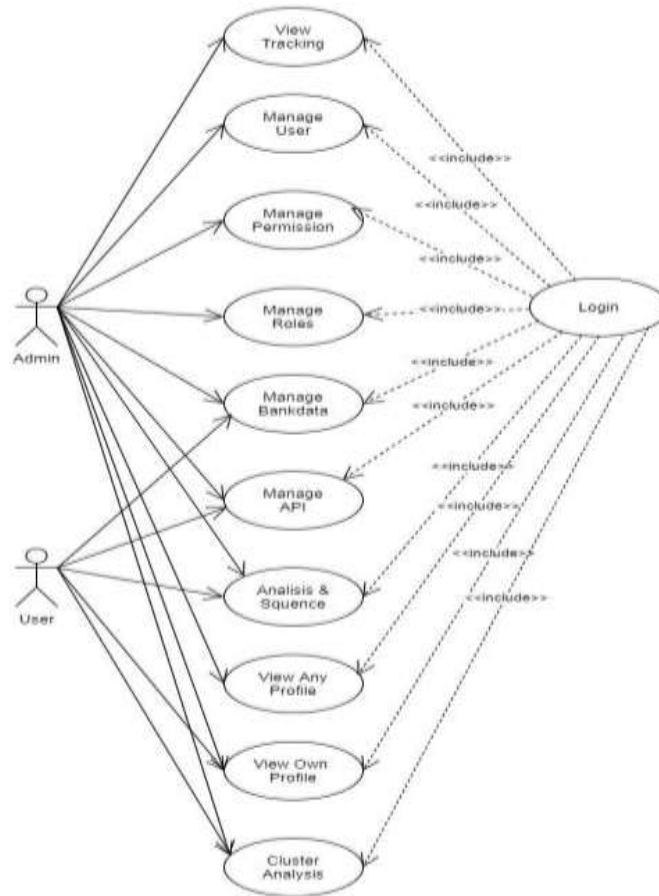
Untuk menjalankan perintah python yang terdapat pada penjelasan di atas digunakan perintah terminal yang terdapat pada server ubuntu 18.04 dengan eksekusi pada tingkat middleware menggunakan Framework Laminas di modul logic. Output yang dihasilkan akan ditampilkan pada presentation layer agar dapat dilihat oleh user baik itu graphic maupun hasil analisisnya dalam bentuk teks

Untuk arsitektur sistem aplikasi dilihat pada gambar sebagai berikut :



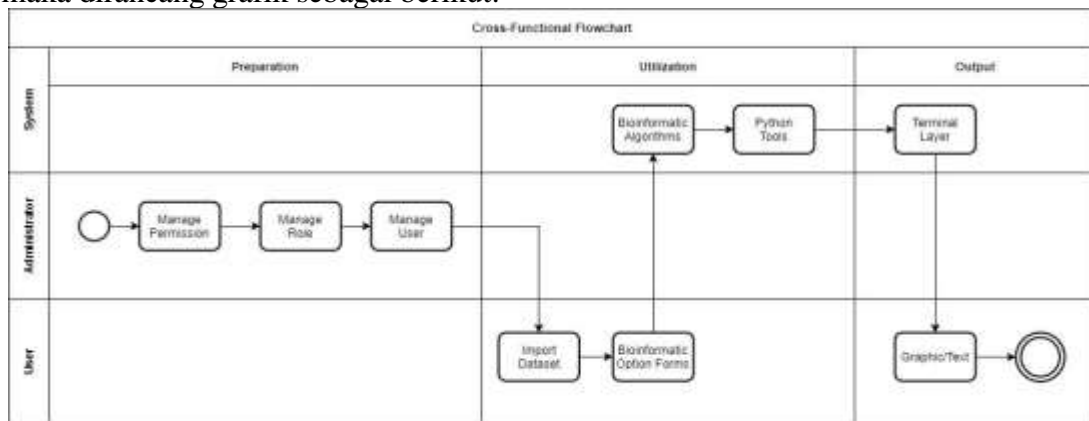
Gambar 8. Rancangan Arsitektur Aplikasi

Pada tingkat bisnis proses aplikasi bioinformatika yang akan dibangun menyesuaikan dengan tingkat kebutuhan yang terdapat pada ilmu pengetahuan bioinformatika sendiri. Umumnya analisis yang terdapat di aplikasi bioinformatika adalah Phenotype Microarray, Cluster Analysis, Sequence Alignment, Sequence, Genome Analysis dan lain lain. Selain itu, dalam aplikasi bioinformatika yang akan dibangun dengan framework laminas ini juga dibangun dengan dua user, admin dan user. Adapun UML use case diagramnya digambarkan secara umum antara lain, sebagai berikut:



Gambar 9. Usecase diagram

Agar lebih memahami bagaimana flow bisnis proses yang terdapat pada aplikasi ini, maka dirancang grafik sebagai berikut:



Gambar 10. Cross-functional Flowchart

Dilihat bahwa proses yang terjadi pada aplikasi berdasarkan penggerak aplikasi bioinformatika yang dirancang dibagi menjadi tiga bagian antara lain user, administrator dan system. System yang dimaksud pada aplikasi ini adalah operating system yang juga ikut andil dalam melaksanakan proses-proses di layer terminal

untuk mengeksekusi perintah-perintah python dalam rangka menggenerate gambar susunan DNA, diagram dan menjalankan dan menangkap keluaran dari pemrograman python.

Pembagian dari sisi pemanfaatan pun terbagi kedalam tiga bagian antara lain Preparation, Utilization dan Output. Preparation berarti proses persiapan agar user dapat menggunakan aplikasi. Proses persiapan ini dilaksanakan oleh administrator yang memiliki hak akses penuh untuk mengatur setiap pengguna. Utilization adalah lapisan pemanfaatan terhadap aplikasi. Karena pemanfaat aplikasi ini adalah user oleh sebab itu user lah yang melakukan import database, mengambil data melalui API dan pemanfaatan perangkat bioinformatic analisis. Terakhir adalah Output. Pada layer output berisi pemrosesan yang dilakukan oleh terminal melalui propagasi yang terdapat pada middleware Framework Laminas berdasarkan data yang diinputkan oleh user pada presentation layer. Setelah output berhasil diperoleh maka tugas disisi middleware adalah menangkap hasil tersebut dan menampilkannya di sisi presentation layer atau view dalam Framework Laminas.

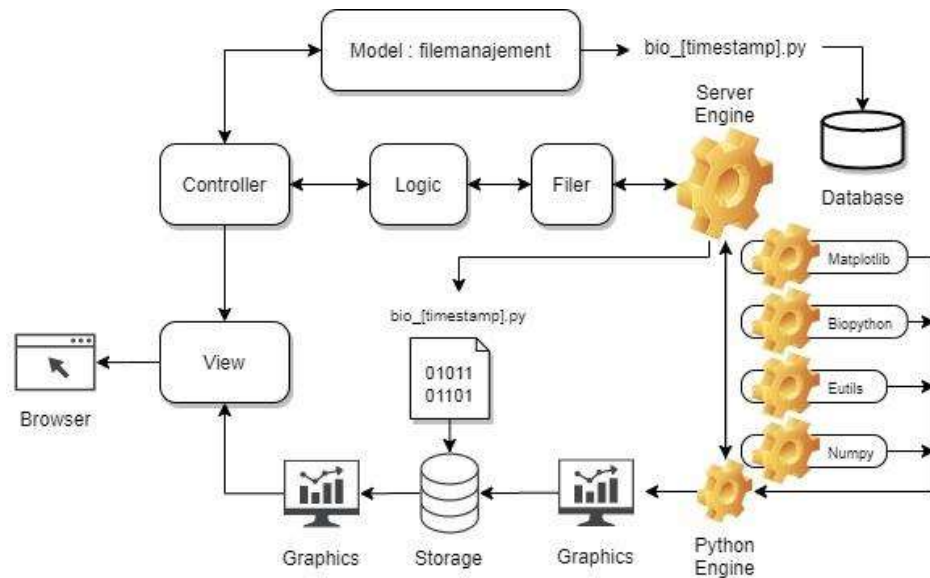
Perancangan Model dan Integrasi ORM

Agar model dapat di integrasikan oleh Doctrine ORM 2(Object Relationship Mapping) model dirancang dengan menggunakan class diagram. Alur prosesnya adalah setelah Doctrine ORM 2 di tambahkan ke dalam library Framework Laminas, setiap model di class dirancang terlebih dahulu kemudian menggunakan terminal di server untuk membuat database sesuai dengan rancangan yang terdapat pada Model.

Rancangan Arsitektur Logic Pada Bisnis Proses

Penambahan modul tambahan dimungkinkan, terlebih agar pemisahan modul setiap bisnis proses dapat dilakukan yang nantinya dapat memudahkan programmer mendefinisikan kesalahan program.

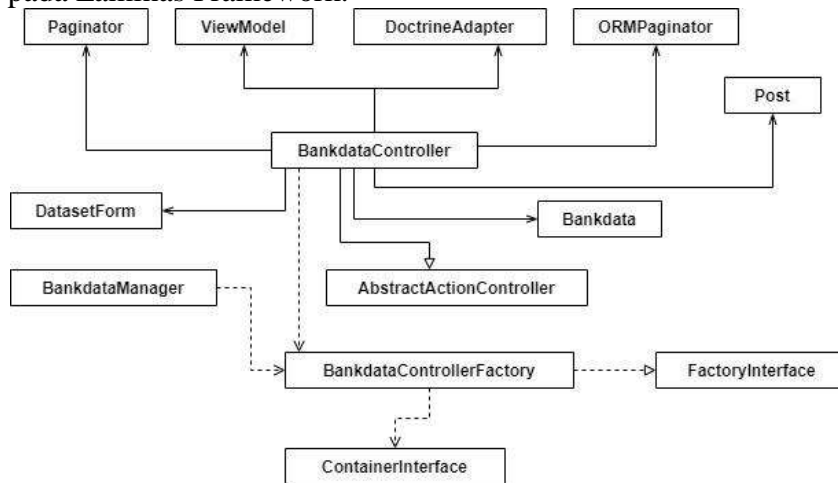
Modul program yang berhubungan dengan console terminal dipisah dan dibuat nama baru menjadi Logic. Nantinya controller akan memanggil Logic dan class yang dituju. Modul logic berisi perintah-perintah untuk melaksanakan algoritma dalam domain ilmu bioinformatika, misalnya : cluster analysis, sequencing analysis, microarray dan lain-lain. Disamping itu di dalam modul logic berisi class yang bertugas membentuk file python dan menjalankan perintah algoritma tersebut kemudian melakukan kompilasi di terminal melalui perintah PHP. Untuk hasil dari perintah kompilasi tersebut di dalam server ditangkap oleh controller dan diteruskan ke view atau presentation layer agar dapat dilihat oleh user



Gambar 11. Rancangan Arsitektur Logic dan Controller

Arsitektur Controller pada Bisnis Proses

Agar controller lebih rapi teratur dan setiap objek yang berisi struktur data dapat dimanfaatkan dengan baik maka perlulah merencanakan class yang terdapat pada controller. Hubungan setiap class yang terdapat pada controller di sesuaikan dengan arsitektur pada Laminas Framework.

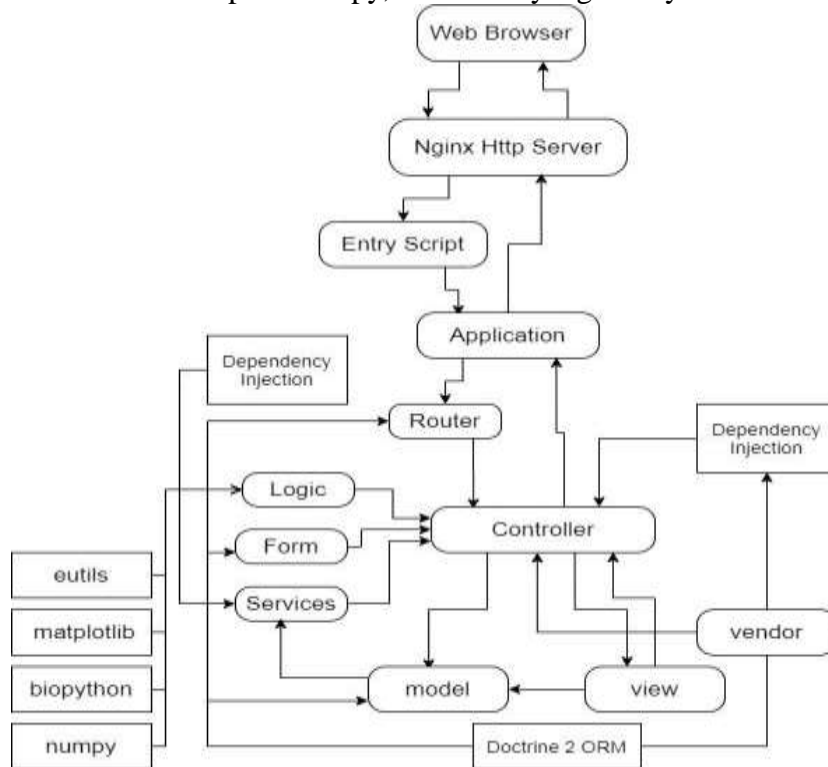


Gambar 11. Rancangan Arsitektur Logic dan Controller

Rancangan Arsitektur Aplikasi Terhadap Integrasi Modul Bioinformatika

Proses integrasi modul-modul python khususnya modul yang berhubungan dalam pemrosesan data bioinformatika dijemantani oleh perintah terminal yang dapat ditrigger melalui perintah PHP. Pemrosesan insilico dilakukan oleh perintah python yang juga mengintegrasikan modul-modul pendukung termasuk biopython, modul

pembentuk gambar dan diagram seperti matplotlib, modul yang memudahkan pemrosesan struktur data seperti numpy, eutils dan yang lainnya



Gambar 12. Arsitektur Aplikasi Terhadap Modul Bioinformatika

Pengujian

Pengujian setiap modul tersebut terdapat beberapa hal yang harus diperhatikan. Sesuai dengan pembahasan dan rancangan pengujian yang terdapat pada gambar 6 Rancangan Pengujian, maka pelaksanaan pengujian dilakukan dengan dua cara terhadap file python yang terbentuk dan satu terhadap Framework Laminas itu sendiri.

Pengujian Pytest

Agar dapat diketahui apakah suatu unit program layak atau tidaknya digunakan maka metode pengujian perangkat lunak dilakukan. Unit testing berfokus pada unit terkecil pada desain perangkat lunak baik itu komponen dan modulnya.

Pada kasus pembuatan aplikasi bioinformatika ini selalu terkait dengan program python sebagai core pemrosesan kasus bioinformatika. Untuk itu pengujian yang dilakukan adalah dengan menguji program python tersebut dengan *plugin* tambahan dari python yang bernama unittest.

Program yang diuji adalah program yang terbentuk dari hasil kompilasi program php yang berisi program python ketika file python selesai dibentuk. Pengujiannya antara lain adalah sebagai berikut.

Ubah file dan fungsi yang akan diuji dengan menambahkan tanda `test_` di awal fungsi dan file.

```

root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python
root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python# ls -l
total 40
drwxr-xr-x 2 root    root    4096 Nov 24 11:48  __pycache__
-rw-r--r-- 1 www-data www-data 609 Oct  7 16:09  clustering.py
-rwxrwxr-x 1 www-data www-data  30 Jul 11 14:17  filename.py
-rw-r--r-- 1 www-data www-data 122 Aug 22 13:50  squence.py
-rw-r--r-- 1 root    root    661 Nov 24 11:26  test_clustering.py
-rw-r--r-- 1 root    root   1038 Nov 23 21:49  test_clustering.pyc
-rw-r--r-- 1 root    root    101 Nov 24 11:22  test_coba.py
-rwxr-xr-x 1 root    root     30 Nov 23 21:40  test_filename.py
-rw-r--r-- 1 root    root    186 Nov 23 21:43  test_filename.pyc
-rw-r--r-- 1 root    root   1788 Nov 24 11:50  test_translate.py
root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python#

```

Gambar 7. Daftar file

Jika program telah dilakukan testing dan terdapat kesalahan maka perbaiki program tersebut sesuai dengan kesalahan yang tampil. Namun jika hasil program tersebut tidak ada kesalahan lagi maka akan muncul tampilan berikut ini.

```

root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python# pytest
===== test session starts =====
platform linux -- Python 3.8.9, pytest-6.1.2, py-3.8.9, pluggy-0.13.1
rootdir: /home/biokiba/ftp/public_html/backoffice/data/python
collected 6 items

test_clustering.py .
test_coba.py .
test_div.py .
test_translate.py .

----- 4 passed in 1.18s -----
root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python#

```

Gambar 8. Pengujian Pytest

Pengujian Unittest

Unittest adalah satu dari beberapa testing di dalam python yang paling umum. Dengan cara melakukan pengecekan satu blok kode untuk memastikan bahwa blok tersebut berjalan dengan baik.

Dalam program python yang telah dibentuk melalui aplikasi ini terdapat diantaranya beberapa blok fungsi yang harus di cek kebenarannya.

```

root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python# python3 -m unittest roman
...
-----
Ran 3 tests in 0.000s

OK
root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python# clear
root@vps727162: /home/biokiba/ftp/public_html/backoffice/data/python# nano roman.py

```

Gambar 9. Unittest

Pengujian Laminas-Test

Agar aplikasi yang dibuat dapat terukur maka aplikasi juga harus diuji. Pengujian juga dilakukan pada Framework Laminas sebagai base framework dimana aplikasi tersebut berada. Untuk pengujian aplikasi, salah satu sampel pengujian adalah pada user controller berikut ini.

Maka hasilnya akan seperti pada gambar dibawah ini:

```
PHPUnit 5.4.6 by Sebastian Bergmann and contributors.

... 3 / 3 (100%)

Time: 116 ms, Memory: 11.00MB

OK (3 tests, 7 assertions)
```

Gambar 4.18 PHP unit testing

PENUTUP

Kesimpulan

Dari keseluruhan proses penelitian yang dilakukan bahwa terdapat tiga permasalahan yang muncul ketika mengimplementasikan Arsitektur Framework Laminas.

1. Pertama adalah permasalahan sumber data bioinformatika yang merupakan raw data, disamping itu permasalahan penyediaan API apakah diproses pada tingkat middleware ataukah hanya diproses di python engine melalui perintah shell. Permasalahan sumber data diselesaikan dengan cara di penyediaan form upload sehingga user dapat melakukan upload data dan dengan cara demikian raw data tersebut dapat diproses. Permasalahan API disetting melalui form penambahan dan konfigurasi data API. Agar memory server tidak overload maka pemrosesan dan pemanggilan data tidak dilakukan di middleware tetapi pada proses di terminal atau perintah shell yang di trigger melalui middleware dengan catatan aplikasi harus terus online selama pemrosesan data.

2. Permasalahan kedua adalah masalah arsitektur aplikasi terhadap bisnis proses. Permasalahan ini membutuhkan pemisahan setiap proses agar developer dapat mudah membagi lapisan proyek dan memudahkan menyelesaikan masalah. Pembuatan Logic pun dilakukan agar proses dari middleware dapat menjadi trigger untuk menjalankan perintah terminal terutama dalam memasukkan setiap algoritma dan analisis bioinformatika kedalam file. Disamping itu class khusus yang menangani file atau dalam konsep ini disebut filer dibuat agar dapat terpisah dengan class algoritmanya.

3. Ketiga adalah permasalahan integrasi modul bioinformatika dengan mengimplementasikan arsitektur aplikasi yang berbasis model, view dan controller ini. Pengintegrasian biopython dan beberapa modul atau library tambahan di dalam library python sendiri sangat bermanfaat dan digunakan untuk menunjang penyelesaian proses insilico. Modul-modul python tersebut diproses dalam layer python engine telah dipisahkan dengan membuat tambahan Logic didalam framework laminas.

Proses pengujian terhadap framework laminas melalui phpunit test yang terdapat pada framework laminas dapat diuji dengan baik dan modul python yang terbentuk diuji dengan pytest dan unittest menampilkan hasil yang baik.

Saran

Berdasarkan pada hasil analisis serta simpulan yang telah diuraikan, maka saran-saran yang dapat di berikan adalah sebagai berikut:

1. Untuk pengembangan lebih lanjut, pentingnya memiliki kapasitas RAM server yang besar sangat dibutuhkan terlebih proses penarikan dan pemrosesan data membuat server down. Penting untuk memperhatikan trafik internet terlebih

pengambilan data melalui API terkadang mencapai lebih dari 100Mb.

DAFTAR PUSTAKA

- Abdurakhmonov, I. Y. (Ed.). (2016). *Bioinformatics: Updated Features and Applications*. BoD–Books on Demand.
- Abdurakhmonov, I. Y. (2016). *Bioinformatics: basics, development, and future*. InTech.
- Bahadur, B., Rajam, M. V., Sahijram, L., & Krishnamurthy, K. V. (Eds.). (2015). *Plant Biology and Biotechnology: Volume II: Plant Genomics and Biotechnology*. Springer India.
- Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Ostell, J., Pruitt, K. D., & Sayers, E. W. (2018). GenBank. *Nucleic acids research*, 46(D1), D41-D47.
- Chang, J. (2020). *Biopython: Tutorial and Cookbook*.
- Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., ... & De Hoon, M. J. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11), 1422-1423.
- Elhakeem, Y. F. G. M., & Barry, B. I. (2013, August). Developing a security model to protect websites from cross-site scripting attacks using ZEND framework application. In 2013 International Conference on Computing, Electrical and Electronic Engineering (Iccee) (pp. 624-629). IEEE.
- Federhen, S. (2012). The NCBI taxonomy database. *Nucleic acids research*, 40(D1), D136-D143.
- Framework.zend.com. (2021). Overview. Diakses 6 Oktober 2021, dari <https://framework.zend.com/about>
- Hamelryck, T., & Manderick, B. (2003). PDB file parser and structure class implemented in Python. *Bioinformatics*, 19(17), 2308-2310.
- Ibrahim, I. M., Abdelmalek, D. H., Elshahat, M. E., & Elfiky, A. A. (2020). COVID-19 spike-host cell receptor GRP78 binding site prediction. *Journal of Infection*, 80(5), 554-562.
- Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., ... & Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10), 1-16.
- Gomez-Casati, D. F., Busi, M. V., Barchiesi, J., Peralta, D. A., Hedin, N., & Bhadauria, V. (2017). Applications of bioinformatics to plant biotechnology. *Current issues in molecular biology*, 27(1), 89-104.
- Goto, N., Prins, P., Nakao, M., Bonnal, R., Aerts, J., & Katayama, T. (2010). BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics*, 26(20), 2617-2619.
- Gupta, O. P., & Rani, S. (2010). Bioinformatics applications and Tools: An Overview. *CiiT-International Journal of Biometrics and bioinformatics*, 3(3), 107-110.

- Lawlor, B., & Walsh, P. (2015). Engineering bioinformatics: building reliability, performance and productivity into bioinformatics software. *Bioengineered*, 6(4), 193-203.
- Lorenz, M., Hesse, G., & Rudolph, J. P. (2016). Object-relational Mapping Revised- A Guideline Review and Consolidation. In ICSoft-EA (pp. 157-168).
- Maulana, S., Cut, B., & Husna, J. (2019). Perancangan Sistem Informasi Transaksi Menggunakan Barcode Zend Berbasis Website. *Kandidat: Jurnal Riset dan Inovasi Pendidikan*, 1(1), 30-35.
- Moitra, D., & Samanta, R. K. (2015). Performance evaluation of BioPerl, Biojava, BioPython, BioRuby and BioSmalltalk for executing bioinformatics tasks. *International Journal of Computer Sciences and Engineering*, 3(01), 157-164.
- Nugroho, E. D., & Rahayu, D. A. (2018). Pengantar Bioteknologi:(Teori dan Aplikasi). Deepublish.
- Pajankar, A. (2017). pytest. In *Python Unit Test Automation* (pp. 87-100). Apress, Berkeley, CA.
- Pratiwi, R. D., Pratiwi, R. H., & Noer, S. (2018). PENINGKATAN KOMPETENSI GURU BIOLOGI MELALUI PENGGUNAAN APLIKASI BIOINFORMATIKA. *Jurnal PKM: Pengabdian kepada Masyarakat* Vol, 1(01).
- Romer, Michael (2016). *Web Development With Zend Framework 3: Concept, Techniques and Practical Solution*, Leanpub
- Schutt, K., & Balci, O. (2016, June). Cloud software development platforms: A comparative overview. In 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA) (pp. 3-13). IEEE.
- Stephenson, L., Wakeham, Y., Seidenman, N., & Choi, J. (2018). Building online genomics applications using BioPyramid. *Bioinformatics*, 34(17), 3055-3057.
- Supaartagorn, C. (2011). PHP Framework for database management based on MVC pattern. *AIRCC's International Journal of Computer Science and Information Technology*, 3(2), 251-258.
- Witarto, A. B. (2003). Bioinformatika: Mengawinkan teknologi informasi dengan bioteknologi. In *Seminar Teknologi Informasi* (pp. 1-6).