

# Sistem Informasi Anotasi Data Penelitian Manajemen Data

Taqarra Rayhan Irfandianto dan Daniel Oranova Siahaan  
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: daniel@if.its.ac.id

**Abstrak**—Anotasi data merupakan proses penandaan, pengaitan, atau pelabelan suatu data. Pada proses anotasi data dapat dilakukan dengan menggunakan Google Colaboratory. Langkah yang dilakukan pada penggunaan Google Colaboratory, yaitu dengan melakukan input data teks dan membuat implementasi *processing* dalam format Python. Akan tetapi, penggunaan implementasi tersebut kurang efektif dan memberikan data hanya dalam satu format. Penggunaan Google Colaboratory juga perlu mengimplementasikan program Python setiap pembuatan *project*. Untuk mengatasi hal tersebut, pembuatan situs web anotasi data merupakan solusi untuk mempermudah pengguna dalam melakukan anotasi data. Situs web anotasi data memiliki fitur berupa penyimpanan database yang berisi program Python yang menjadi implementasi dalam anotasi data tersebut. Program Python tersebut mencakup penggunaan *Natural Language Processing* (NLP) dalam dilakukan *processing data* yang akan dianotasi. Melalui implementasi tersebut akan mempermudah peneliti dalam melakukan proses anotasi data. Situs web ini berfokus pada anotasi data menggunakan *framework* React.js sebagai implementasi *front-end* dan Flask sebagai implementasi *back-end*. Pada *framework* React.js memiliki beberapa fitur seperti export data output dengan format yang ditentukan. Selain itu, pada sisi backend juga dilakukan pembuatan API. Program Python akan disimpan dalam database menggunakan API upload file. Melalui situs web ini diharapkan dapat mempermudah peneliti dalam melakukan proses anotasi data dan melakukan pengolahan data *output*.

**Kata Kunci**—Anotasi data, API, *Natural Language Processing*, Python Flask, React.js.

## I. PENDAHULUAN

KECERDASAN buatan atau *Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang dikhususkan untuk memecahkan masalah kognitif yang umumnya terkait dengan kecerdasan manusia, seperti pembelajaran, pemecahan masalah, dan pengenalan pola. Pada implementasi AI diperlukan data untuk dijadikan data testing dalam percobaan untuk mendapatkan hasil sesuai [1]. Pada implementasi AI digunakan *machine learning* dalam pemrosesannya. *Machine learning* bertujuan untuk melakukan proses otomatis terhadap model analitik untuk melakukan tugas kognitif seperti deteksi objek alami [2].

Salah satu bidang AI, yaitu *Natural Language Processing* (NLP). NLP adalah kemampuan program komputer untuk memahami bahasa manusia dalam melakukan penafsiran terhadap bahasa manusia [3]. Contoh implementasi dalam NLP adalah proses anotasi data. Anotasi data merupakan suatu pelabelan atau penandaan pada sebuah informasi yang digunakan untuk penelitian dan pengelolaan suatu dataset. *Sentiment analysis* merupakan salah satu dari implementasi anotasi data. *Sentiment analysis* merupakan suatu proses penafsiran terhadap suatu komentar atau pos berupa teks di mana memiliki nilai yang mencakup data penafsiran yang diteliti [4]. Pada anotasi data dibutuhkan dataset sebagai data

yang akan diuji untuk mendapat sistem yang sesuai dengan kebutuhan peneliti. Dataset ini digunakan sebagai penampung data dalam tabel tunggal di mana setiap kolomnya mewakili nilai tertentu. Proses anotasi data dapat dilakukan melalui Google Colaboratory dengan menggunakan Python sebagai bahasa untuk melakukan implementasi. Pada umumnya, penggunaan anotasi dilakukan oleh anotator dengan melakukan implementasi *preprocess* dan *processing* terhadap dataset yang dimasukkan. Pada implementasi tersebut hanya bisa dilakukan satu kali untuk setiap dataset yang digunakan. Dari implementasi anotasi tersebut juga memiliki kekurangan dalam fleksibilitas penggunaannya. Selain itu, proses tersebut hanya bisa dilakukan oleh anotator yang memahami mengenai pelabelan. Hal tersebut mengakibatkan peneliti memerlukan waktu yang lebih lama dalam melakukan proses pelabelan data dan mendapatkan hasil statistik berdasarkan dataset yang diterima.

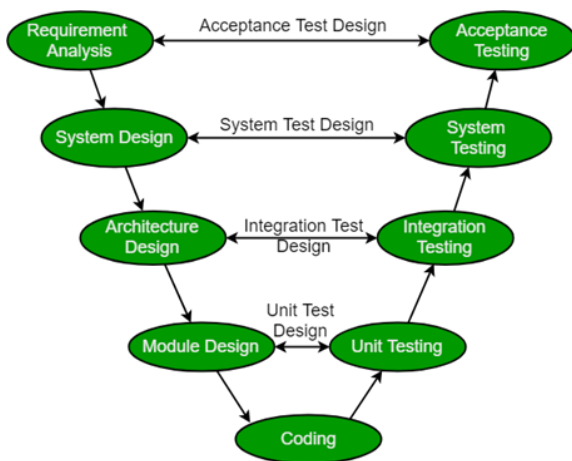
Berdasarkan masalah tersebut, penelitian ini akan dibangun aplikasi berbasis situs web yang menyediakan anotasi data dengan program atau keinginan anotator. Aplikasi Anotasi Data Penelitian (ADP) memiliki fitur dalam melakukan custom anotasi menggunakan program *library* Python dengan melakukan input terhadap data model tersebut. Python memiliki kelebihan di mana dapat melakukan pengembangan web, Analisa data, pemrosesan data dalam satu proyek [5] Hasil dari penelitian ini berupa aplikasi untuk Pengembangan Sistem Informasi Anotasi Data Penelitian. Aplikasi ini diharapkan dapat mempermudah pengguna dalam melakukan anotasi data sesuai dengan dataset yang diberikan dan model yang diunggah anotator dan mendapatkan hasil berupa data terstruktur dari penelitian yang dijalankan.

## II. TINJAUAN PUSTAKA

Tinjauan Pustaka menjelaskan terkait berbagai teori yang digunakan sebagai dasar dalam pembuatan aplikasi pengembangan sistem informasi anotasi data penelitian. Selain itu, terdapat penjelasan mengenai gambaran secara umum sistem informasi yang akan dibangun.

### A. Anotasi Data

Anotasi data merupakan suatu proses yang digunakan untuk melakukan identifikasi terhadap suatu teks untuk mendapatkan suatu hasil berupa informasi tambahan. Tujuan dari anotasi data adalah pelabelan dari suatu teks yang mudah dipahami oleh sistem. Penelitian yang digunakan dalam implementasi situs web anotasi data, yaitu penelitian *Sentiment analysis* [6]. Penelitian ini melakukan anotasi data menggunakan data target *positive*, dan *negative*. Dataset yang digunakan dalam penelitian berjumlah 400 data. Proses anotasi data dalam penelitian memiliki tahapan dalam melakukan prediksi.



Gambar 1. Struktur proses V-Model.

### B. Python Flask

Python Flask adalah sebuah *framework web micro* yang disediakan Python diperuntukkan oleh para pengembang perangkat lunak maupun pengelolaan situs web. Flask memiliki struktur yang sederhana, fleksibel, dan mudah dipelajari. Itu didasarkan pada pustaka WSGI Werkzeug dan *engine template* Jinja, serta menggunakan aturan rute dan dekorator Python untuk mengelola permintaan HTTP.

Pada implementasi Python Flask digunakan sebagai *framework back-end* dalam pembuatan API. Penggunaan Python Flask memiliki keuntungan dalam implementasi pemanggilan *library* dan melakukan *processing* terhadap data yang akan dilakukan anotasi. untuk menyimpan data user dan program Python dalam bentuk Python *file* [7].

### C. React JS

React.js adalah front-end JavaScript *library* yang efisien dan fleksibel untuk pengembangan antarmuka pengguna. React.js dikelola oleh Facebook sebagai JavaScript *library* sumber terbuka yang dapat digunakan untuk membuat *views* yang dijalankan di atas HTML. React memiliki beberapa fitur utama yang menjadikannya lebih unggul dibandingkan dengan *library* JavaScript lainnya seperti JSX, DOM Virtual, komponen *property react*, dan keuntungan fitur lainnya. ReactJS pada dasarnya memungkinkan pengembangan aplikasi berbasis web yang besar dan kompleks yang dapat mengubah datanya tanpa harus *me-refresh* halaman berikutnya [8].

Pada implementasi digunakan react.js untuk pembuatan situs web dinamis. Setiap fitur dapat dilakukan modifikasi terhadap *request* yang akan dikirim ke *database* melalui API *backend*. Pada react.js terdapat beberapa fitur seperti *export* hasil data dalam format yang diinginkan.

### D. V-Model

*Software Development Life Cycle (SDLC)* adalah proses yang menghasilkan perangkat lunak dengan kualitas tertinggi dalam waktu sesingkat mungkin atau bisa disebut pembuatan proyek dengan menggunakan metode yang efektif [9]. Salah satu model pengembangan perangkat lunak yang digunakan adalah V-Model. V-Model merupakan perpanjangan dari model *Waterfall* di mana proses eksekusi model mengalir dalam “V-shape”. Model-V menentukan serangkaian fase yang dijalankan secara berurutan, satu per satu hingga proyek selesai. Model ini juga dikenal sebagai “model Verifikasi dan

Validasi” [10]. Penggunaan V-Model ini berstruktur seperti V dan memiliki proses di setiap *step*-nya. Ada 9 tahap pada V-Model yang dibagi menjadi dua, yaitu pada fase verifikasi dan fase validasi seperti pada Gambar 1.

### E. Natural Language Processing (NLP)

*Natural Language Processing (NLP)* adalah cabang dari AI yang berhubungan dengan interaksi antara mesin dan manusia menggunakan bahasa natural. NLP ini digunakan dalam *processing* situs web anotasi data dengan menggunakan bahasa Python. Program Python yang diimplementasikan akan dimasukkan dalam *database* menggunakan *back-end* Python Flask [11].

Proses anotasi data dengan NLP yang akan digunakan yaitu mengenai pelabelan suatu kalimat. Pelabelan ini disebut dengan anotasi data. Suatu teks dikelompokkan ke dalam suatu kategori dari konten tersebut [12].

### F. Use Case Alistair CockBurn

*Use case* adalah salah satu dokumentasi dengan menggambarkan alur interaksi pada suatu sistem yang dibuat. *Use case* berfungsi dalam memberi penjelasan detail dalam suatu kasus sistem dari perangkat lunak. Salah satu *use case* yang digunakan adalah Alistair CockBurn. *Use case* ini berfokus pada interaksi pada pengguna dengan sistem yang digunakan [13].

### G. Requirement Traceability Matrix

*Requirement Traceability Matrix* adalah salah satu dokumen yang digunakan untuk melacak hubungan antara kebutuhan fungsional, desain sistem, dan kasus pengujian. Penggunaan *traceability matrix* dilakukan dengan menghubungkan 2 data pengembangan perangkat lunak. Kebutuhan akan ditinjau secara detail untuk memastikan bahwa mereka konsisten, benar, dan lengkap sepanjang siklus pengembangan [14].

## III. METODOLOGI

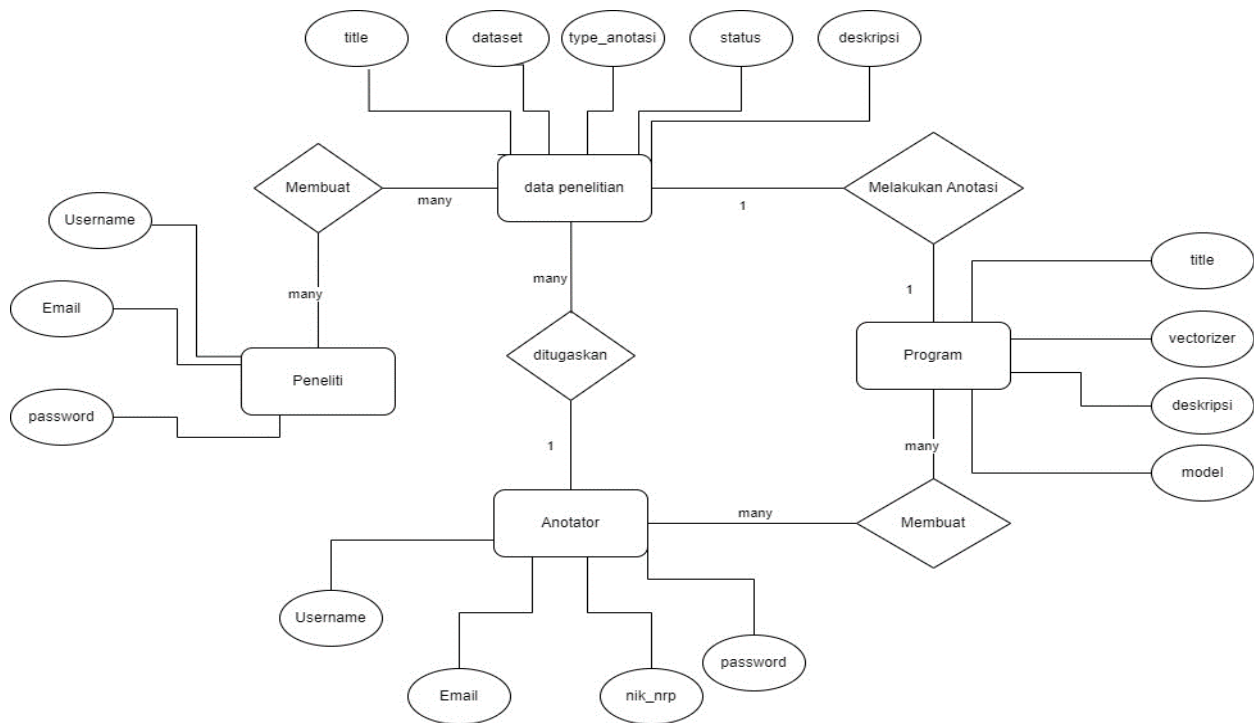
Tinjauan Pustaka menjelaskan terkait berbagai teori yang digunakan sebagai dasar dalam pembuatan aplikasi pengembangan sistem informasi anotasi data penelitian. Selain itu, terdapat penjelasan mengenai gambaran secara umum sistem informasi yang akan dibangun.

Pada V-Model ini digunakan dua fase, yaitu fase verifikasi dan validasi. Pada fase verifikasi penelitian akan dilakukan beberapa tahapan sebelum dilakukan implementasi.

### A. Analisis Kebutuhan

Proses dilakukan dengan menggunakan metode penelitian kualitatif. Metode kualitatif ini digunakan untuk menyelidiki, menemukan, menggambarkan, dan menjelaskan kebutuhan dari situs web anotasi data.

Proses secara *asynchronous* dilakukan melalui Google Meet dengan mahasiswa. Survei yang dilakukan menggunakan survei longitudinal di mana dilakukan untuk jangka yang panjang hingga proses pembuatan situs web selesai. Survei ini dilakukan dengan harapan situs web yang diimplementasikan memiliki hasil yang maksimal untuk *client*. Beberapa pertanyaan yang menjadi acuan dalam survei tersebut sebagai berikut: (1) Apa kebutuhan yang diinginkan dalam pembuatan anotasi data? (2) Apa fitur yang dibutuhkan dalam proses



Gambar 2. ERD situs web anotasi data.

anotasi data? (3) Bagaimana alur proses anotasi data menggunakan situs web yang diinginkan? (4) Bagaimana proses anotasi data menggunakan program Python secara manual? (5) Adakah fitur tambahan yang diperlukan dalam situs web ini? Kemudian dari langkah-langkah yang dilakukan, luaran yang diharapkan berupa data kebutuhan utama, kebutuhan fungsional, dan kebutuhan non-fungsional dari implementasi. Data tersebut disertai dengan spesifikasi kasus untuk setiap kebutuhan.

### B. Desain Sistem

Proses desain sistem merupakan proses lanjutan dari analisis kebutuhan untuk membentuk suatu sistem dari sebuah situs web. Desain sistem diawali dengan pembuatan *use case diagram* untuk memahami alur pembuatan sistem yang akan dibuat. *Use case diagram* digunakan dalam pembuatan *Entity Relationship Diagram* (ERD). ERD ini menjadi acuan dalam pembuatan tabel dalam situs web. Proses desain sistem diakhiri dengan pembuatan rancangan unit sistem. Rancangan ini digunakan sebagai tolak ukur pengujian sistem.

Proses ini dilakukan selama 2 minggu dan mendapatkan hasil berupa *use case diagram*, ERD, dan rancangan pengujian sistem dari implementasi. Tahap pengujian sistem dilakukan di fase validasi ditahap pengujian sistem.

### C. Desain Arsitektur

Proses ini memberikan perincian pada sistem yang akan dibuat. Dalam desain arsitektur ini digunakan *two-tier arsitektur*. Arsitektur ini merupakan pendekatan di mana aplikasi dibagi menjadi 2 bagian yaitu *frontend (client)* dan *backend (server)*.

### D. Desain Modul

Desain modul dilakukan dengan identifikasi tingkat rendah seperti implementasi modul yang lebih spesifik. Tahap ini dilakukan perincian mengenai setiap implementasi yang

diberikan. Penjelasan algoritma pada implementasi dirancang menggunakan *sequence diagram*. Pada tahap ini diberikan rancangan *input* dan *output* pada setiap komponen. Proses ini berlangsung sekitar 1 minggu. Tujuan dari pengujian ini memastikan setiap unit berfungsi sesuai dengan keinginan fitur.

### E. Implementasi

Pembuatan implementasi dilakukan setelah mendapatkan data kebutuhan situs web. Tahap ini dilakukan sampai selesai untuk masuk pada tahap validasi. Alokasi waktu yang ditentukan menyesuaikan dengan pembuatan fitur yang dikerjakan. Pembuatan implementasi diperkirakan kurang lebih 1 bulan. Dalam melakukan implementasi perlu memperhatikan data rancangan yang sudah dibuat untuk mempermudah dalam proses pengembangan situs web.

### F. Pengujian Unit

Pengujian unit atau unit testing dilakukan dalam jangka waktu kurang lebih 2 minggu. Tahap ini dilakukan pengujian terhadap dilakukan pengujian pada bagian *frontend* maupun *backend*.

### G. Pengujian Integrasi

Pengujian integrasi dilakukan dengan menyambungkan sisi *backend* dengan *frontend*. Pengujian ini berfungsi untuk melakukan validasi integrasi di situs web sudah sesuai dengan keinginan pembuatan fitur.

### H. Pengujian Sistem

Pengujian sistem dilakukan sebagai validasi setiap sistem yang telah diimplementasikan. Proses pengujian dilakukan dengan cara pengujian terhadap semua fitur yang sudah dilakukan *unit testing* dan *integration testing*. Pengujian sistem ini dilakukan dalam jangka waktu 2 minggu dengan *output* penggunaan fungsi dan fitur sesuai dengan rancangan implementasi unit.

Tabel 1.  
Hasil Pengujian Unit

ID Test	Deskripsi	Hasil Penguji
UT01	test_register_success	Berhasil
UT02	test_register_failed	Berhasil
UT03	test_login_success	Berhasil
UT04	test_login_failed	Berhasil
UT05	test_profile_success	Berhasil
UT06	test_profile_failed	Berhasil
UT07	test_update_profile_success	Berhasil
UT08	test_update_profile_duplicate	Berhasil
UT09	test_update_profile_not_found	Berhasil
UT10	test_update_penelitian_success	Berhasil
UT11	test_update_penelitian_no_file	Berhasil
UT12	test_add_anotators	Berhasil
UT13	test_add_anotators_failed	Berhasil
UT14	test_get_penelitian_success	Berhasil
UT15	test_get_my_penelitian_success	Berhasil
UT16	test_edit_penelitian_success	Berhasil
UT17	test_edit_penelitian_failed	Berhasil
UT18	test_update_role_success	Berhasil
UT19	test_get_data_progress_success	Berhasil
UT20	test_get_data_finished	Berhasil

Tabel 2.  
Hasil Pengujian Integrasi

ID Test	Deskripsi	Hasil Penguji Integrasi
UT01	Penelitian dapat melakukan akun	Berhasil
UT02	Penelitian dapat melakukan login akun yang sudah terdaftar	Berhasil
UT03	User (peneliti, admin, dan anotator) dapat logout akun	Berhasil
UT04	User dapat melihat detail akun	Berhasil
UT05	User dapat melihat update akun	Berhasil
UT06	User dapat melakukan update role ke anotator	Berhasil
UT07	Peneliti dapat membuat penelitian	Berhasil
UT08	Peneliti dapat menambahkan anotator dalam penelitian	Berhasil
UT09	Anotator dapat melakukan pembuatan program model	Berhasil
UT10	Anotator melihat data program model	Berhasil
UT11	Anotator melihat data program model yang dibuat	Berhasil
UT12	Anotator melakukan update data program model yang dibuat	Berhasil
UT13	Anotator melakukan delete data Program model yang dibuat	Berhasil
UT14	Anotator dapat melakukan anotasi otomatis	Berhasil
UT15	Anotator dapat melakukan anotasi secara manual	Berhasil
UT16	Anotator dapat melakukan update status penelitian	Berhasil
UT17	User dapat menampilkan progres anotasi	Berhasil
UT18	User dapat menampilkan hasil anotasi	Berhasil
UT19	User dapat melakukan download terhadap hasil anotasi	Berhasil
UT20	Admin menambahkan update di situs web	Berhasil
UT21	Admin dapat melakukan update user (peneliti dan anotator)	Berhasil
UT22	Admin dapat melihat detail user (peneliti dan anotator)	Berhasil
UT23	Admin dapat melakukan delete user (peneliti dan anotator)	Berhasil

I. Pengujian Penerimaan

Tahap akhir dalam Model-V di mana dilakukan pengujian langsung oleh peneliti sebagai *client*. Pelaksanaan ini dilakukan secara *online* melalui Zoom Meeting. *Output* berupa *feedback* dari peneliti mengenai situs web yang digunakan dan dilakukan perencanaan kembali jika terdapat hal yang perlu diperbaiki atau dikembangkan.

IV. HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai langkah-langkah yang dilaksanakan dalam analisis kebutuhan, desain sistem, desain arsitektur, desain modul, implementasi. Hasil dari pengujian ditampilkan dalam pengujian unit, pengujian integrasi, pengujian sistem, dan *acceptance testing*.

A. Analisis Kebutuhan

Proses analisa diawali dengan melakukan wawancara dengan pengguna. Melalui wawancara tersebut didapatkan data kebutuhan utama dari pengguna. Kebutuhan utama tersebut dipecah menjadi kebutuhan fungsional dan non-fungsional untuk mempermudah dalam memahami kebutuhan sebelum dilakukan perancangan. Setelah didapatkan kebutuhan fungsional dan non-fungsional, proses selanjutnya yaitu membuat spesifikasi kebutuhan untuk menjelaskan secara rinci mengenai kebutuhan yang dimaksud.

B. Desain Sistem

Desain sistem memerlukan data berupa identifikasi kasus yang didapatkan melalui analisis kebutuhan. Pembuatan *use case diagram* juga memperhatikan hubungan dari setiap tokoh yang akan dibuat. Proses dilanjutkan dengan membuat *entity relationship diagram* (ERD) melalui *use case* yang tersedia. Gambar 2 menunjukkan hasil ERD yang dirancang berdasarkan *use case* yang tersedia.

C. Desain Arsitektur

*Use Case Diagram* yang didapatkan melalui desain sistem memberi pemahaman mengenai interaksi antara pengguna dengan sistem. Sedangkan, ERD memberi pandangan mengenai aliran sistem yang akan dibuat. Data tersebut memberikan kesimpulan berupa *Data Flow Diagram* (DFD).

*Data Flow Diagram* pada situs web Anotasi Data memiliki 2 level. Perancangan tersebut memberikan informasi mengenai entitas yang diperlukan dalam pembuatan situs web seperti *user*, penelitian, dan program model untuk melakukan anotasi. Setelah melalui proses tersebut, pembuatan entitas dapat dilakukan dengan mengidentifikasi alur dan kebutuhan dalam arsitektur.

D. Desain Modul

Tahap desain modul memiliki hasil berupa perincian unit untuk implementasi. Desain modul memiliki tahap validasi yaitu tahap pengujian unit. Proses ini diawali melalui analisa desain arsitektur menjadi data yang lebih mudah dipahami alurnya melalui pembuatan *Sequence Diagram*. Pembuatan *sequence diagram* memiliki tujuan berupa pemberian gambaran alur setiap sistem berjalan.

E. Implementasi

Implementasi dilakukan sesuai dengan rancangan yang sudah dibuat pada fase verifikasi. Implementasi memiliki 4 bagian penting yaitu: (1) Pembuatan HTTP *request frontend* menggunakan React. (2) Pembuatan fungsi penerima data penelitian. (3) *Preproccssing* data menggunakan flask. (4) Pembuatan *output* data penelitian.

Setelah tahap implementasi dilakukan tahap validasi. Da-

Tabel 3. Perbedaan Pengelolaan Anotasi Tanpa Meng

No.	Deskripsi Fitur	Manajemen Anotasi Tanpa Meng Web Anotasi Data
1.	Anotasi data manual	(1) Anotator melakukan anotasi aplikasi seperti Excel. (2) Peneliti tidak dapat melihat langsung.
2.	Anotasi otomatis oleh anotator	(1) Peneliti memberikan dataset anotator. (2) Anotator membuat implemen Google Collaboratory. (3) Anotator memasukkan dataset (4) Anotator melakukan <i>runn</i> dataset. (5) Anotator memberikan hasil un
3.	Manajemen penelitian	Tidak memiliki daftar data peneliti
4.	Manajemen implementasi	Satu implementasi digunakan untu
ST12	yang dibuat Anotator melakukan <i>update</i> data program model yang dibuat	Berhasil
ST13	Anotator melakukan <i>delete</i> data Program model yang dibuat	Berhasil
ST14	Anotator dapat melakukan anotasi otomatis	Berhasil
ST15	Anotator dapat melakukan anotasi secara manual	Berhasil
ST16	Anotator dapat melakukan <i>update</i> status penelitian	Berhasil

lam tahap validasi juga memerlukan implementasi jika terjadi *bug* dalam proses pengujian.

F. Pengujian Unit

Pengujian unit dilakukan untuk memeriksa implementasi pada unit terkecil. Proses ini menggunakan *tools* javascript yaitu *JEST* untuk melakukan pengecekan terhadap komponen tampilan yang sudah diimplementasikan. Hasil pengujian dibuat dalam dokumentasi di Tabel 1.

Tabel 1 memberikan hasil pengujian dengan hasil indeks pengujian berhasil di atas 90%. Dengan hasil tersebut, proses dapat dilanjutkan pada pengujian integrasi.

G. Pengujian Integrasi

Pengujian integrasi dilakukan untuk memastikan setiap implementasi tampilan tersambung dengan *database*. Pengujian ini dilakukan dengan melakukan pengecekan fitur di bagian tampilan situs web.

Berdasarkan hasil pengujian integrasi di Tabel 2, didapatkan *index* keberhasilan lebih dari 90%. Hasil tersebut membuktikan bahwa sistem sudah terintegrasi dengan baik dan memberikan respons *database* sesuai dengan rancangan yang sudah dibuat.

H. Pengujian Sistem

Pengujian sistem merupakan tahap validasi sebelum dilakukan pengujian langsung dengan pengguna. Pengujian ini bertujuan untuk memastikan sistem yang dibangun sudah memenuhi kriteria kebutuhan pengguna. Tabel 3 memberikan bukti mengenai hasil pengujian sistem. Semua pengujian yang dilakukan memiliki hasil pengujian berhasil. Melalui tahap tersebut, sistem dapat dilanjutkan pada tahap pengujian langsung dengan pengguna.

I. Pengujian Penerimaan

Pengujian tahap akhir ini dilakukan langsung oleh peneliti yang terkait, yaitu mahasiswa S3. Dari tahap ini dilakukan

Tabel 4. Keterangan Penggunaan

Situs web Anotasi Data	Versi: 1.0
<i>User Acceptance Testing</i>	Tanggal: 18 Juni 2023
UAT 1.0	

Tabel 5. Hasil *User Acceptance Testing*

ID	Deskripsi Pengujian	Hasil Penguji
UAT01	Penelitian dapat melakukan pembuatan akun	Berhasil
UAT02	Penelitian dapat melakukan <i>login</i> pada akun yang sudah terdaftar	Berhasil
UAT03	<i>User</i> membuat penelitian	Berhasil
UAT04	<i>User</i> dapat <i>update</i> akun	Berhasil
UAT05	<i>User</i> dapat melakukan <i>update</i> <i>role</i> ke anotator	Berhasil
UAT06	Anotator dapat melakukan pembuatan program model	Berhasil
UAT07	Anotator dapat melakukan anotasi otomatis	Berhasil
UAT08	Anotator dapat melakukan anotasi secara manual	Berhasil
UAT09	Peneliti dapat melihat data progres anotasi milik anotator	Berhasil
UAT10	<i>User</i> (peneliti, anotator, dan admin) dapat melakukan <i>download</i> hasil penelitian	Berhasil
UAT11	Admin dapat melakukan <i>update</i> data notasi	Berhasil
UAT12	Admin dapat melakukan <i>delete</i> data notasi	Berhasil
UAT13	Admin dapat melakukan <i>update</i> <i>user</i> (peneliti dan anotator)	Berhasil
UAT14	Admin dapat melakukan <i>delete</i> <i>user</i> (peneliti dan anotator)	Berhasil
UAT15	Admin dapat melakukan <i>update</i> data program model	Berhasil
UAT16	Admin dapat melakukan <i>delete</i> pada data program model	Berhasil

proses demo dengan menggunakan situs web anotasi data seperti Tabel 4 secara langsung oleh peneliti dalam melakukan proses anotasi. Tabel 5 diberikan penjelasan mengenai pengujian langsung oleh *client*. Pengujian tersebut sudah memenuhi kebutuhan pengguna dan situs web siap digunakan secara umum.

J. Pembahasan

Penerapan pembuatan situs web anotasi data menggunakan metode V-model menunjukkan pentingnya melakukan perancangan situs web sebelum dilakukan implementasi. Penerapan juga memerlukan jadwal yang efektif sehingga memiliki hasil yang maksimal. Dalam penerapan memerlukan fase verifikasi di mana melakukan analisa dan perancangan mengenai implementasi yang akan dibuat. Setelah implementasi dibuat, fase validasi dibutuhkan untuk memastikan situs web berjalan dengan baik dan memenuhi kriteria pengguna.

Berdasarkan dari perbandingan pengalaman pengguna melakukan anotasi. Situs web anotasi memiliki kelebihan dalam melakukan proses anotasi lebih dari satu proyek. Sistem memiliki fitur untuk melakukan *upload* terhadap program model yang akan digunakan. Tabel 6 menunjukkan perbedaan mengenai pengelolaan anotasi tanpa menggunakan situs web dan menggunakan situs web.

Melalui perbandingan kualitatif tersebut, memberikan hasil bahwa situs web anotasi data dapat membantu mahasiswa dalam melakukan pengelolaan anotasi yang lebih

efisien dan mudah. Penggunaan implementasi juga tidak memerlukan waktu lama jika dibandingkan dengan penge-lolaan anotasi data secara manual.

## V. KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan dan saran yang dapat diberikan mengenai perangkat lunak yang dikembangkan dalam penelitian ini. Kesimpulan dirancang berdasarkan hasil pembuatan situs web anotasi data. Saran dirancang melalui pembahasan yang didapatkan selama pengembangan perangkat lunak dan diberikan untuk pengembangan selanjutnya dalam penelitian ini.

Situs web anotasi data merupakan situs web untuk melakukan proses anotasi data. Proses anotasi data membutuhkan suatu program model untuk melakukan prediksi. Penggunaan Python merupakan salah satu solusi dalam implementasi anotasi data. Proses *cleaning* data dilakukan pada input data penelitian yang kemudian nantinya akan dilakukan anotasi secara otomatis. *Export* hasil penelitian dapat dilakukan dengan memberi hasil anotasi data dalam format json, dan hasil tersebut dapat diubah dalam bentuk yang lebih variatif pada bagian *frontend* seperti dalam bentuk *pie chart* atau *bar chart*. Data yang dalam format json dapat digunakan untuk *export file* dalam bentuk *.csv*. Dalam perbandingan dengan Google Colaboratory, situs web ini menawarkan keunggulan dalam aspek manajemen penelitian. Keunggulan tersebut terletak pada kemampuan program model yang dapat diintegrasikan ke dalam *database* dan digunakan secara berkelanjutan tanpa adanya batasan yang membatasi.

Terdapat beberapa saran dari implementasi yang sudah dilaksanakan dalam penelitian ini. Harapannya dengan adanya saran yang diberikan, penelitian ini menjadi lebih baik dari sebelumnya. Saran yang dapat diberikan dari implementasi program model dapat dibuat lebih variatif sehingga dapat mencakup lebih luas penggunaan anotasi data.

## DAFTAR PUSTAKA

- [1] E. Adetiba *et al.*, "Evolution of artificial intelligence programming languages - a systematic literature review," *J. Comput. Sci.*, vol. 17, no. 11, 2021, doi: 10.3844/jcssp.2021.1157.1171.

- [2] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.
- [3] S. Pais, J. Cordeiro, and M. L. Jamil, "NLP-based platform as a service: a brief review," *J. Big Data*, vol. 9, no. 1, p. 54, 2022, doi: 10.1186/s40537-022-00603-5.
- [4] H. T. Phan, V. C. Tran, N. T. Nguyen, and D. Hwang, "Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model," *IEEE Access*, vol. 8, pp. 14630–14641, 2020, doi: 10.1109/ACCESS.2019.2963702.
- [5] N. Nandal, R. Tanwar, and A.-S. K. Pathan, "Sentiment Analysis based Emotion Extraction for COVID-19 Using Crawled Tweets and Global Statistics for Mental Health," in *Procedia Computer Science*, 2023, vol. 218, pp. 949–958. doi: 10.1016/j.procs.2023.01.075.
- [6] E. Asmawati, A. Saikhu, and D. Siahaan, "Sentiment Analysis of Text Memes: A Comparison among Supervised Machine Learning Methods," in *Proceedings - 9th International Conference on Electrical Engineering, Computer Science and Informatics, EECSI 2022*, 2022, pp. 349–354. doi: 10.23919/EECSI56542.2022.9946506.
- [7] E. Susanti, "Implementasi restful api dalam pembuatan master data planogram menggunakan framework flask (studi kasus: pt sumber alfaría trijaya, tbk)," *J. Teknol. Inf.*, vol. 19, no. 3, pp. 295–307, 2020, doi: 10.33633/tc.v19i3.3468.
- [8] M. Kaluža and B. Vukelić, "Comparison of front-end frameworks for web applications development," *Zb. Veleučilišta u Rijeci*, vol. 6, no. 1, pp. 261–282, Jan. 2018, doi: 10.31784/zvr.6.1.19.
- [9] O. E. Olorunshola and F. N. Ogwueleka, "Review of System Development Life Cycle (SDLC) Models for Effective Application Delivery," in *Proc. Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, 2021, pp. 281–289. doi: 10.1007/978-981-16-0739-4\_28.
- [10] J. E. T. Akinsola, A. S. Ogunbanwo, O. J. Okesola, I. J. Odun-Ayo, F. D. Ayegbusi, and A. A. Adebisi, "Comparative Analysis of Software Development Life Cycle Models (SDLC)," in *Proc. Advances in Intelligent Systems and Computing*, 2020, vol. 1224 AISC, pp. 310–322. doi: 10.1007/978-3-030-51965-0\_27.
- [11] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.
- [12] A. Junnarkar, S. Adhikari, J. Faganía, P. Chimurkar, and D. Karia, "E-mail Spam Classification Via Machine Learning and Natural Language Processing," in *Proc. of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, (ICICV)*, 2021, pp. 693–699. doi: 10.1109/ICICV50876.2021.9388530.
- [13] M. Ben-Menachem, "Writing effective use cases," *ACM SIGSOFT Softw. Eng. Notes*, vol. 26, no. 1, pp. 94–95, Jan. 2001, doi: 10.1145/505894.505918.
- [14] H. P. Putro and A. F. Wibowo, "Software verification and validation on object oriented software development using traceability matrix," in *Proc. of the 3rd International Conference on Informatics and Computing, ICIC 2018*, 2018, pp. 1–5. doi: 10.1109/IAC.2018.8780518.