

**RANCANG BANGUN APLIKASI WEB PENDETEKSI WARNA PADA
*PIXEL GAMBAR DENGAN KNN CLASSIFIER***

(Skripsi)

Oleh

KADAFI EKA SAKTI



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

**RANCANG BANGUN APLIKASI WEB PENDETEKSI WARNA PADA
PIXEL GAMBAR DENGAN *KNN CLASSIFIER***

Oleh

KADAFI EKA SAKTI

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

ABSTRAK

RANCANG BANGUN APLIKASI WEB PENDETEKSI WARNA PADA *PIXEL* GAMBAR DENGAN *KNN CLASSIFIER*

Oleh

KADAFI EKA SAKTI

Manusia memiliki keterbatasan dalam mendeteksi ratusan warna dalam gambar secara presisi. Banyak orang awam mengalami kesulitan dalam mengklasifikasikan sebuah warna pada benda, pada umumnya orang awam hanya mengetahui warna dasar atau warna yang umum. Sehingga aplikasi web bernama DYSCO ini dibuat untuk mempermudah dalam mengklasifikasikan sebuah warna. DYSCO merupakan aplikasi pendeteksi warna berbasis *web* dengan menggunakan metode *K-Nearest Neighbors*. Aplikasi web ini dibuat dengan menggunakan bahasa pemrograman *Python* dan kerangka kerja mikro *Flask*. Aplikasi web ini dapat menampilkan informasi warna *pixel* suatu gambar yang di-*upload* pengguna ke dalam *web*, dengan mengekstraksi fitur *RGB* pada gambar tersebut. Metode pengembangan yang digunakan adalah *AI Project Cycle* dengan 6 tahapan yaitu *Problem Scoping, Data Acquisition, Data Exploration, Modelling, Evaluation dan Deployment*. Pembuatan aplikasi web pendeteksi warna pada *pixel* ini telah berhasil dilakukan dan sudah mampu mendeteksi dan memberikan informasi warna yang di klik pengguna pada gambar, dengan akurasi yang tinggi yaitu 0,844. Aplikasi web juga sudah dapat memenuhi fungsi utamanya, ditunjukkan melalui pengujian menggunakan teknik *Usability Testing* berdasarkan ISO 9421-11 untuk mengukur *Efectivity, Eficiency dan Satisfaction*, dengan 10 orang responden awam. Hasil *Usability Testing* mendapatkan skor yang sangat bagus dengan nilai *efectivity* yaitu 100% dengan predikat sangat efektif, nilai *eficiency* yaitu 32,6 detik dengan predikat sangat cepat, dan nilai *satisfaction* yaitu 84 dengan predikat dapat diterima (*acceptable*).

Kata Kunci : *K-Nearest Neighbor, AI Project Cycle, Flask, Usability Testing*

ABSTRACT

DESIGN AND DEVELOPMENT OF COLOR DETECTION WEB APPLICATION IN IMAGE PIXEL USING KNN CLASSIFIER

By

Kadafi Eka Sakti

Humans have limitation in detecting the hundreds of colors in an image with precision. Many ordinary people have difficulty classifying a color in objects, in general, ordinary people only know basic colors or common colors. So a web application called DYSCO was made to make it easier to classify a color. DYSCO is a web-based color detection application using the K-Nearest Neighbors method. This web application is built using the Python programming language and the Flask micro framework. This web application can display pixel color information of an image that the user uploads to the web, by extracting the RGB features of the image. The development method used is the AI Project Cycle with 6 stages, namely Problem Scoping, Data Acquisition, Data Exploration, Modeling, Evaluation and Deployment. The creation of this pixel color detection web application has been successfully carried out and has been able to detect and provide color information that user clicks on the image, with a high accuracy of 0.844. The web application has also been able to fulfill its main functions, shown through testing using the Usability Testing technique based on ISO 9421-11 to measure Effectiveness, Efficiency and Satisfaction, with 10 respondents. Usability Testing results get a very good score with an effectiveness value of 100% with a very effective predicate, an efficiency score of 32.6 seconds with a very fast predicate, and a satisfaction score of 84 with an acceptable predicate.

Keywords : *K-Nearest Neighbor, AI Project Cycle, Flask, Usability Testing*

RIWAYAT HIDUP



Penulis dilahirkan di Sungailiat pada tanggal 29 April 1999. Penulis merupakan anak pertama dari 2 bersaudara dari pasangan Bapak Muhammad Kaidi , S.E. dan Ibu Ida Safrina , S.Pd..

Penulis memulai jenjang pendidikan dari TK Stannia Muntok, SD Negeri 22 Muntok dan lulus pada tahun 2011, SMP Negeri 1 Muntok dan lulus pada tahun 2014, SMA Negeri 1 Muntok dan lulus pada tahun 2017 dan ditahun yang sama diterima pada Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN.

Selama menjalani proses perkuliahan secara aktif, penulis mengikuti Himpunan Mahasiswa Teknik Elektro (HIMATRO) sebagai Sekretaris Departemen Divisi Komunikasi dan Informasi pada Periode 2018 dan menjadi Sekretaris Umum Pengurus Harian HIMATRO Periode 2019. Pada organisasi luar kampus, penulis juga mengikuti komunitas Gerakan Digital Ekosistem Nusantara (GRADIEN). Selain proses perkuliahan, penulis juga pernah melaksanakan kerja praktik di Dinas Komunikasi dan Informasi Pemerintahan Daerah Bangka Barat pada tahun 2020 yang bertempat di Kabupaten Bangka Barat, Provinsi Bangka Belitung, dan masuk dalam Bidang Penyelenggaraan *E-Government*. Kemudian pada bulan Juli 2022 hingga Februari 2023 penulis berhasil menyelesaikan penelitian Tugas Akhir dengan judul penelitian **“Rancang Bangun Aplikasi Web Pendeteksi Warna Pada Pixel Gambar Dengan *Knn Classifier*”**.

Judul Skripsi

: **RANCANG BANGUN APLIKASI WEB
PENDETEKSI WARNA PADA *PIXEL*
GAMBAR DENGAN *KNN CLASSIFIER***

Nama Mahasiswa

: **Kadafi Eka Sakti**

Nomor Pokok Mahasiswa

: **1715061022**

Program Studi

: **Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

: **Teknik**



1. Komisi Pembimbing

Dr. Eng. Mardiana, S.T., M.T.
NIP 19720316 199903 2 002

Rio Ariestia Pradipta, S.Kom, M.T.I.
NIP 19860323 201903 1 013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Herlinawati, S.T., M.T.
NIP. 19710314 199903 2 001

Ketua Program Studi
Teknik Informatika

Mona Arif Muda, S.T., M.T.
NIP 19711112 200003 1 002

Judul Skripsi

: **RANCANG BANGUN APLIKASI WEB
PENDETEKSI WARNA PADA *PIXEL*
GAMBAR DENGAN *KNN CLASSIFIER***

Nama Mahasiswa

: **Kadafi Eka Sakti**

Nomor Pokok Mahasiswa

: **1715061022**

Program Studi

: **Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

: **Teknik**



1. Komisi Pembimbing

Dr. Eng. Mardiana, S.T., M.T.
NIP 19720316 199903 2 002

Rio Ariestia Pradipta, S.Kom, M.T.I.
NIP 19860323 201903 1 013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

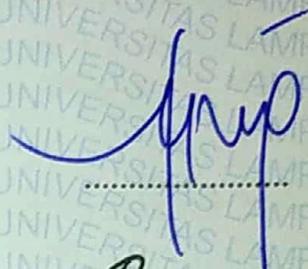
Herlinawati, S.T., M.T.
NIP. 19710314 199903 2 001

Mona Arif Muda, S.T., M.T.
NIP 19711112 200003 1 002

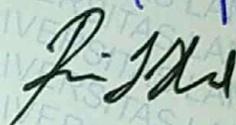
MENGESAHKAN

1. Tim Penguji

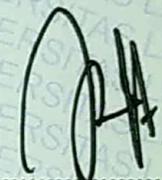
Ketua : **Dr. Eng. Mardiana, S.T., M.T.**



Sekretaris : **Rio Ariestia Pradipta, S.Kom, M.T.I.**



Penguji : **Yessi Mulyani, S.T., M.T.**



2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. ✎
NIP. 19750928 200112 1 002



Tanggal Lulus Ujian Skripsi : 06 Maret 2023

“Dan nikmat Tuhan mana lagi yang kamu dustakan.”

(Q.S. Ar-Rahman)

“Tidak masalah jika kamu berjalan dengan lambat, asalkan kamu tidak pernah berhenti berusaha.”

(Confucius)

“Saat kamu bersantai-santai ada jutaan bahkan miliaran orang yang mencoba mengambil posisimu kelak dimasa depan.”

“Hidup jangan sesuai tuntutan”

(Luthfi halimawan)

"I'm so happy that you're not afraid to be who you are. Because that's one of the most important things a person has to overcome."

(Jacksepticeye)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang. *Alhamdulillah* dengan segala rahmat dan karunia-Nya, skripsi ini dapat diselesaikan dengan baik dan lancar. Semoga dengan keberhasilan yang telah dicapai ini saya dapat menuju masa depan yang lebih baik dan dapat menggapai cita-cita serta selalu berada di jalan-Mu.

KUPERSEMBAHKAN KARYA ILMIAH INI
TERUNTUK:

“Ibunda Ida Safrina dan Muhammad Kaidi atas dukungan dan kasih sayang yang diberikan mulai dari saya ada di dunia ini sampai saya sudah besar seperti sekarang ini. Tak lupa terima kasih pula atas kebahagiaan-kebahagian yang diberikan selama ini. Semoga ini menjadi tahap awal dalam meraih kesuksesan sehingga dapat membahagiakan Ibu dan Bapak”

“Diriku sendiri. Maaf untuk malam-malam panjang dengan mata yang sulit tertidur, kepala yang sakit, dan lelah pikiran. Terima Kasih telah berjuang dan bekerjasama selama ini. Kamu hebat.”

“Terima Kasih kepada teman-teman Teknik Informatika 2017 yang telah menemani dan membantu saya selama perkuliahan di kampus tercinta Universitas Lampung. Terima Kasih atas kenangan yang kalian berikan selama perkuliahan mulai dari proses pengenalan kampus sampai dengan akhir semester. Semoga kelak kita semua akan menjadi orang-orang yang sukses.”

“Terima Kasih kepada teman-teman lainnya yang selalu memberikan semangat dan motivasi saat saya jatuh dan yang selalu mengingatkan saya untuk bersyukur saat saya bahagia. Semoga Allah selalu memberikan kita kebahagiaan di apapun jalan yang kita pilih kedepannya.”

SANWACANA

Puji syukur penulis panjatkan kepada Allah *سُبْحَانَهُ وَ تَعَالَى*, yang telah memberikan karunia serta ridho-Nya sehingga penulis dapat melaksanakan dan menyelesaikan penelitian ini yang berjudul “Rancang Bangun Aplikasi Web Pendeteksi Warna Pada *Pixel* Gambar Dengan *Knn Classifier*”. Penelitian ini merupakan salah satu sarat untuk menyelesaikan kurikulum mata kuliah penelitian skripsi pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Pelaksanaan penelitian ini penulis mendapatkan bantuan, bimbingan serta pengarahan dari berbagai pihak. Maka dari itu, penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Allah *سُبْحَانَهُ وَ تَعَالَى* yang senantiasa memberikan kemudahan dan kelancaran kepada penulis serta Rasulullah Muhammad *صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ* yang telah menjadi suri tauladan selama penelitian berlangsung;
2. Ibu dan Ayah serta keluarga penulis yang selalu memberikan motivasi dan dukungan kepada penulis;
3. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung;
4. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
5. Bapak Mona Arif Muda, S.T.,M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan telah membantu proses kelancaran pengerjaan penelitian;

6. Ibu Dr. Eng. Mardiana, S.T., M.T., selaku Pembimbing Utama penelitian yang selalu meluangkan waktunya untuk memberikan bimbingan dan dukungan;
7. Bapak Rio Ariestia Pradipta, S.Kom, M.T.I., selaku Pembimbing Pendamping penelitian yang selalu memberikan motivasi dan memberikan bimbingan kepada penulis untuk menjadi lebih baik;
8. Ibu Yessi Mulyani, S.T.,M.T., selaku Penguji penelitian yang telah banyak memberikan saran dan masukan;
9. Bapak Ing. Hery Dian Septama, S.T.selaku Pembimbing Akademik yang telah memberikan bimbingan selama perkuliahan disetiap semester dan selalu memberikan motivasi;
10. Mbak Rika selaku Admin Program Studi Teknik Informatika yang telah memberikan bantuan dalam proses administrasi penelitian;
11. Teman-teman Teknik Informatika 2017 yang selalu mendukung penulis;
12. Semua pihak yang turut serta dalam membantu menyelesaikan penelitian dan tidak mungkin penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan penelitian ini masih bisa disempurnakan kembali. Oleh karena itu, penulis mengharapkan saran dan kritik yang bersifat membangun dari para pembaca. Penulis berharap laporan skripsi ini dapat bermanfaat bagi banyak pihak.

Bandar Lampung, 14 Februari 2023

Penulis,



Kadafi Eka Sakti

DAFTAR ISI

	Halaman
DAFTAR ISI.....	i
DAFTAR TABEL.....	iii
DAFTAR GAMBAR.....	iv
I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
II. TINJAUAN PUSTAKA.....	5
2.1 Gambaran Umum Perusahaan.....	5
2.2 <i>Artificial Intelligence</i>	7
2.3 <i>Python</i>	8
2.4 <i>AI Project Life Cycle</i>	9
2.5 Aplikasi Web.....	11
2.6 <i>Flask</i>	12
2.7 <i>HTML</i>	14
2.8 <i>CSS</i>	16
2.9 <i>JavaScript</i>	18
2.10 <i>Visual Studio Code</i>	19
2.11 <i>Anaconda</i>	20
2.12 <i>Google Collab</i>	21
2.13 <i>K-Nearest Neighbor</i>	22
2.14 <i>HEROKU</i>	23
2.15 <i>Usability Testing</i>	24
2.16 <i>Trello</i>	30
2.17 <i>Piksel</i>	31
2.18 <i>RGB</i>	32
2.20 <i>Diagram Komponen</i>	33
2.19 <i>Penelitian Terkait</i>	35
III. METODE PENELITIAN.....	41
3.1 Waktu dan Tempat Penelitian.....	41

3.1.1	Jadwal Penelitian.....	41
3.2	Alat Dalam Penelitian.....	42
3.2.1	Alat Penelitian.....	42
3.2.2	Bahan Penelitian.....	43
3.3	Tahapan Penelitian.....	43
3.3.1	Tahap <i>Problem Scoping</i>	43
3.3.2	Tahap <i>Data Acquisition</i>	45
3.3.3	Tahap <i>Data Exploration</i>	46
3.3.4	Tahap <i>Modelling</i>	48
3.3.5	Tahap Evaluation.....	49
3.3.6	Tahap Deployment.....	49
IV.	PEMBAHASAN.....	51
4.1	Pembahasan.....	51
4.1.1	<i>Data Acquisition</i> dan <i>Data Exploration</i>	51
4.1.2	<i>Modelling</i>	52
4.1.3	<i>Evaluation</i>	61
4.1.4	<i>Deployment</i>	64
4.1.5	Pengujian <i>Usability</i>	75
4.1.6	Analisis Keterbatasan.....	79
4.1.7	Capaian Penelitian.....	79
V.	SIMPULAN DAN SARAN.....	81
5.1	Simpulan.....	81
5.2	Saran.....	81
	DAFTAR PUSTAKA.....	82

DAFTAR TABEL

Tabel	Halaman
Tabel 2. 1 Standar Ukuran Efektifitas [27]	26
Tabel 2. 2 Standar Ukuran Efisiensi [28].....	27
Tabel 3. 1 Jadwal Penelitian.....	41
Tabel 3. 2 Alat yang Digunakan Dalam Penelitian.....	42
Tabel 4. 1 Daftar Task.....	76
Tabel 4. 2 Hasil Pengukuran Efektifitas	76
Tabel 4. 3 Hasil Pengukuran Efisiensi	77
Tabel 4. 4 Perhitungan Skor SUS	78

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2. 1 Logo Orbit Future Academy	5
Gambar 2. 2 Domain Artificial Intelligence [22].....	7
Gambar 2. 3 Contoh Kode Python	8
Gambar 2. 4 Alur Kerja AI Project Life Cycle [1]	9
Gambar 2. 5 Contoh Aplikasi Web.	11
Gambar 2. 6 Contoh Penerapan Flask.....	13
Gambar 2. 7 Contoh Script HTML.	15
Gambar 2. 8 Contoh Script CSS.	17
Gambar 2. 9 Contoh Script JavaScript.....	18
Gambar 2. 10 Antarmuka IDE Visual Studio Code.....	19
Gambar 2. 11 Tampilan UI Anaconda Navigator.	20
Gambar 2. 12 Tampilan UI Google Collab.....	21
Gambar 2. 13 Ilustrasi Klasifikasi K-NN [23].....	22
Gambar 2. 14 Rumus Euclidean Distance 1 Variabel [23]	23
Gambar 2. 15 Rumus Euclidean Distance n variabel [23].....	23
Gambar 2. 16 Tampilan UI HEROKU.....	24
Gambar 2. 17 Rumus Penilaian Efektifitas [29]	26
Gambar 2. 18 Rumus Perhitungan Efisiensi [29].....	27
Gambar 2. 19 Pertanyaan SUS [19]	28
Gambar 2. 20 Chart Nilai Pertanyaan SUS [18]	28
Gambar 2. 21 Rumus Perhitungan SUS [18]	29
Gambar 2. 22 Grafik Kesimpulan SUS [18]	30
Gambar 2. 23 Tampilan Board Trello	30
Gambar 2. 24 Tampilan Pikel pada Berbagai Display [30].....	31
Gambar 2. 25 Representasi nilai RGB pada Gambar [33].....	33
Gambar 2. 26 Component	34
Gambar 2. 27 Dependencies	34
Gambar 2. 28 Interfaces	34
Gambar 3. 1 Tahapan Penelitian.	43
Gambar 3. 2 Contoh Data Pada Website Kaggle.....	46
Gambar 3. 3 Contoh Dataset	47
Gambar 3. 4 Contoh Visualisasi Data	47

Gambar 3. 5 Klasifikasi dengan metode K-NN [23].....	48
Gambar 3. 6 Contoh Hasil Akhir Tampilan	49
Gambar 4. 1 Dataset Warna dan nilai RGB	52
Gambar 4. 2 Contoh Dataset Awal	53
Gambar 4. 3 Hasil Perhitungan Jarak.....	54
Gambar 4. 4 Penentuan Jarak Terdekat.....	54
Gambar 4. 5 Penentuan Kelas dengan K.....	55
Gambar 4. 6 Fitur Inspect Web Browser	56
Gambar 4. 7 Nilai RGB pada Gambar	56
Gambar 4. 8 Gambar Percobaan	57
Gambar 4. 9 Gambar percobaan dengan Colour Picker.....	57
Gambar 4. 10 Gambar Percobaan pada Aplikasi Web DYSCO	58
Gambar 4. 11 Nilai RGB Zaffre.....	58
Gambar 4. 12 Library Pemodelan	59
Gambar 4. 13 Input Data Frame.....	59
Gambar 4. 14 Convert Dataframe ke Format Numpy	60
Gambar 4. 15 Import KNN Classifier	60
Gambar 4. 16 Model KNN.....	61
Gambar 4. 17 Library Evaluasi	61
Gambar 4. 18 Perhitungan Accuracy, Precision dan Recall	62
Gambar 4. 19 Perhitungan n Terbaik	63
Gambar 4. 20 Baris Kode Python dengan Flask	65
Gambar 4. 21 Baris kode JavaScript.....	66
Gambar 4. 22 Baris Kode HTML	67
Gambar 4. 23 Loading Server Flask pada Anaconda Prompt.....	68
Gambar 4. 24 Tampilan Halaman Web pada Browser	69
Gambar 4. 25 Activity Diagram.....	70
Gambar 4. 26 Backlog Perancangan Aplikasi.....	71
Gambar 4. 27 Nama dan Logo Aplikasi.....	72
Gambar 4. 28 Laman Bagian Pertama	72
Gambar 4. 29 Laman Bagian Kedua	73
Gambar 4. 30 Laman Bagian Ketiga.....	73
Gambar 4. 31 Button Upload	73
Gambar 4. 32 Hasil Warna.....	74
Gambar 4. 33 Laman Setelah Deteksi Warna	74
Gambar 4. 34 Diagram Komponen Aplikasi Web DYSCO	75

I. PENDAHULUAN

1.1 Latar Belakang

Pada zaman yang semakin berkembang, teknologi merupakan suatu hal yang sangat penting. Dunia saat ini tengah menghadapi era teknologi revolusi industri 4.0 dimana kondisi tersebut ditandai langsung dengan penggunaan mesin digital dan internet. Semua kemajuan dan perubahan yang dibawa revolusi industri 4.0 mungkin membuat banyak orang merasa tidak ada revolusi lagi yang dapat terjadi. Namun, pikiran itu musnah setelah muncul era *Society 5.0* yang merupakan konsep oleh federasi bisnis Jepang. *Era Society 5.0* bertujuan untuk mengintegrasikan ruang maya dan ruang fisik. Era transformasi digital mengubah kondisi bisnis dan sosial sehingga mengharuskan masyarakat untuk beradaptasi dengan perkembangan teknologi.

Salah satu bidang teknologi yang menjadi penyumbang utama dalam transformasi digital adalah bidang *Artificial Intelligence* (AI). AI merupakan pengembangan dan integrasi dari bidang elektronika, ilmu komputer, dan matematika. Sistem AI dibuat untuk dapat melakukan pekerjaan manusia seperti berpikir, mengambil keputusan, dan melakukan klasifikasi. Tuntutan keterampilan transformatif harus diiringi dengan adanya pendidikan di masyarakat. Pendidikan tersebut dapat dilakukan melalui program yang diadakan oleh pemerintah maupun instansi yang memiliki tenaga ahli di dalamnya.

Salah satu program yang dicanangkan oleh pemerintah yang bermitra dengan perusahaan yaitu program Magang dan Studi Independen Bersertifikat

(MSIB). Program tersebut menjembatani masyarakat khususnya mahasiswa dalam memperoleh edukasi terkait perkembangan teknologi di perusahaan. Hal ini menjadi kesempatan besar untuk mahasiswa agar dapat melatih keterampilan transformatif mereka. Oleh karena itu, menjadi salah satu alasan mengikuti program MSIB terkhusus program yang diselenggarakan oleh *Orbit Future Academy* yaitu program Studi Independen Bersertifikat (SIB) “*Foundation of AI and Life Skills for Gen Z*”. Program tersebut merupakan program pelatihan AI secara *online* untuk pelajar yang bertujuan bukan hanya untuk memperkenalkan teknologi AI pada pelajar, tetapi juga untuk memungkinkan mereka bisa mengangkat perangkat AI sehingga mampu membuat produk yang menciptakan dampak sosial. Dalam membuat produk untuk penelitian akhir, fokus pada penelitian ini adalah domain *Computer Vision (CV)*. Berdasarkan sub bab I.1, dilakukan pembuatan penelitian “Aplikasi Web Pendeteksi Warna pada *Pixel* Gambar dengan *KNN Classifier*”.

Klasifikasi sebuah gambar dalam suatu label tertentu adalah salah satu hal yang sangat mudah bagi manusia karena pada hakikatnya manusia merupakan makhluk visual yang bisa melihat, mendeteksi serta mengklasifikasi bentuk data baik data berupa teks ataupun data gambar yang dikolektifkan menjadi sebuah *dataset* yang dapat dianalisis. Namun, pada dasarnya manusia juga memiliki keterbatasan dalam mengklasifikasi suatu hal. Pada penelitian akhir ini, dengan membawa latar belakang masalah klasifikasi warna berdasarkan *pixel* gambar, hal ini dikarenakan manusia memiliki keterbatasan dalam mengklasifikasi dan mendeteksi warna warna yang ada pada gambar. *Pixel* itu sendiri berunsur dari 3 warna yaitu hijau, biru, dan merah (RGB) dan dari setiap *inch* warna itu dapat menghasilkan 16 juta warna dari perubahan atau penambahan *inch* tersebut . Oleh karena itu, solusi dari permasalahan ini adalah menggunakan bantuan metode AI yaitu *K-Nearest Neighbor (K-NN)* dalam mengidentifikasi warna-warna tersebut menjadi suatu label nama warna yang presisi.

K-NN adalah salah satu metode AI yang merupakan metode klasifikasi yang sangat sederhana dalam mengklasifikasikan sebuah gambar atau data berdasarkan jarak terdekat dengan tetangganya. Data yang memiliki jarak fitur *vector* terdekat akan menjadi satu kelas atau label klasifikasi. Hasil dari sampel uji

yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN. Jarak yang digunakan adalah jarak *Euclidean Distance*. Walaupun sangat sederhana, namun K-NN ini mudah dipahami

Dalam penelitian ini dilakukan pembuatan sebuah aplikasi web dengan kode nama DYSCO yang dapat digunakan untuk mengidentifikasi warna dari gambar. Tujuan pembuatan aplikasi web ini adalah untuk membantu dalam kesulitan pengenalan warna dalam sebuah gambar yang dapat dialami oleh pengguna, pengguna disini adalah seseorang yang mengidap buta warna ataupun orang awam yang ingin tahu suatu warna dari sebuah gambar. Pengguna dapat mengunggah gambar ke situs web dan mengklasifikasi warna dalam gambar sesuai keinginan Pengguna dengan cara mengeklik cursor pada warna dalam gambar untuk mengklasifikasikan dan mendeteksi warna dari sebuah gambar. Informasi warna yang didapat dapat digunakan pengguna untuk berbagai hal, bagi pengguna pengidap buta warna informasi ini dapat digunakan untuk membandingkan dengan warna lain, untuk pengguna lain informasi ini dapat digunakan untuk berbagai bidang seperti desain yang membutuhkan informasi akurat mengenai warna. Ruang warna utama yang digunakan dalam penelitian ini adalah RGB (*Red, Green, Blue*). Aplikasi web dibangun menggunakan *Flask*. Implementasi dari penelitian ini adalah dengan menggunakan algoritma K-NN, KNN dipilih sebagai algoritma dikarenakan berdasarkan beberapa penelitian disimpulkan bahwa KNN dapat digunakan dalam klasifikasi warna dengan tingkat akurasi yang tinggi lebih dari 90% (0,9), KNN dilatih dengan nilai histogram warna R, G, dan B, warna dari citra masukan yang diberikan akan dideteksi dan kemudian dievaluasi modelnya serta dapat menghasilkan *output* berupa nama warna sesuai dengan metode klasifikasi K-NN.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu bagaimana membangun sebuah aplikasi berbasis web pengenalan warna pada gambar yang memungkinkan pengguna mengetahui warna *pixel* dari gambar yang diinginkan menggunakan algoritma *artificial intelligence* dan KNN (*K-nearest Neighbor*).

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membangun sebuah aplikasi berbasis web pengenalan warna pada gambar yang memungkinkan pengguna mengetahui warna *pixel* dari gambar yang diinginkan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat dalam mempermudah pengguna untuk mengetahui warna dalam suatu gambar.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Membahas mengenai pengembangan Aplikasi Web Pendeteksi Warna Pada *Pixel* Gambar Dengan *KNN Classifier* yang diimplementasikan pada *web browser*.
2. Aplikasi yang dikembangkan masih bersifat lokal *server* dan belum diimplementasikan secara publik.

II. TINJAUAN PUSTAKA

2.1 Gambaran Umum Perusahaan



Gambar 2. 1 Logo *Orbit Future Academy*

Gambar 2.1 merupakan logo dari *Orbit Future Academy*. *Orbit Future Academy* (OFA) didirikan pada tahun 2016 dengan tujuan untuk meningkatkan kualitas hidup melalui inovasi, edukasi, dan pelatihan keterampilan. Label atau *brand* Orbit merupakan kelanjutan dari warisan mendiang Prof. Dr. Ing. B. J. Habibie (presiden Republik Indonesia ke-3) dan istrinya, Dr. Hasri Ainun Habibie. Mereka berdua telah menjadi penggerak dalam mendukung perkembangan inovasi dan teknologi pendidikan di Indonesia. OFA mengkurasi dan melokalkan program/kursus internasional untuk *upskilling* atau *reskilling* pemuda dan tenaga kerja menuju pekerjaan masa depan. Hal ini sesuai dengan slogan OFA, yakni “*Skills-for-Future-Jobs*”.

Visi:

Memberikan pembelajaran berbasis keterampilan transformatif terbaik untuk para pencari kerja & pencipta lapangan kerja.

Misi:

1. Membangun jaringan *Orbit Transformation Center* (OTC) secara nasional untuk menyampaikan kurikulum keterampilan masa depan berbasis sertifikasi melalui Platform Konten Digital.
2. Secara proaktif bekerja dengan pemerintah & organisasi dengan mengubah tenaga kerja mereka agar sesuai dengan perubahan pekerjaan yang terjadi karena Industri 4.0.
3. Melatih pemuda dengan keterampilan kewirausahaan & mencocokkan mereka dengan peluang masa depan yang muncul di berbagai industri.
4. Menghubungkan jaringan inkubator dan akselerator yang dikurasi ke industri, investor, dan ekosistem *start-up* global.

Berikut adalah deskripsi pekerjaan student sebelum pengerjaan PA:

- a. Mengikuti *pre-test*.
- b. Mengikuti kelas sesi pagi pada pukul 08.00 hingga 11.30 WIB.
- c. Mengikuti kelas sesi siang pada pukul 13.00 hingga 16.30 WIB.
- d. Mengulang materi yang telah disampaikan di kelas sesi pagi dan siang, setelah kelas sesi siang, selama 1 jam (*self-study*).
- e. Mengerjakan latihan individu atau kelompok yang diberikan oleh *homeroom* atau *domain coach* saat kelas berlangsung.
- f. Mengerjakan tugas yang diberikan *homeroom* atau *domain coach* hingga batas waktu tertentu.
- g. Mengerjakan *mini project* yang diberikan *homeroom* atau *domain coach* hingga batas waktu tertentu
- h. Mengikuti *post-test*.

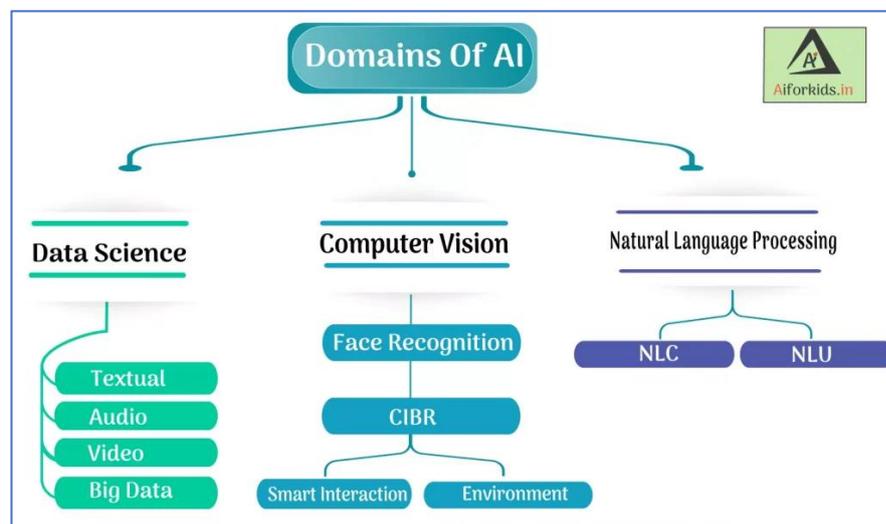
Student memiliki peran sebagai *AI Engineer* selama pengerjaan PA, dengan deskripsi pekerjaan sebagai berikut:

- a. Bertanggung jawab dalam pemilihan model, algoritma dan metode AI.
- b. Bertanggung jawab dalam *deployment project* (*backend*).
- c. Membuat dan menguji model serta mencocokkan dengan data.

- d. Membuat program *Python* menggunakan kerangka kerja *Flask* (*web-based*) untuk *deployment project*.
- e. Mengawasi proses *deployment* ke *web hosting* (HEROKU).

2.2 Artificial Intelligence

Kecerdasan buatan atau *Artificial Intelligence* (AI) mengacu pada simulasi kecerdasan manusia dalam mesin yang diprogram untuk berpikir dan bertindak seperti manusia. Istilah ini juga dapat diterapkan pada mesin apa pun yang menunjukkan fungsi yang terkait dengan pikiran manusia [1].



Gambar 2. 2 Domain *Artificial Intelligence* [22]

Gambar 2.2 menunjukkan 3 domain utama dalam *Artificial Intelligence* yaitu *Data Science*, *Computer Vision* dan *Natural Language Processing*. Karakteristik ideal kecerdasan buatan adalah kemampuannya untuk merampingkan dan melakukan tindakan yang paling mungkin untuk mencapai tujuan tertentu. Bagian dari kecerdasan buatan adalah *Machine Learning* (ML). ML mengacu pada konsep bahwa program komputer dapat secara otomatis belajar dan beradaptasi dengan data baru tanpa bantuan manusia. Teknik pembelajaran mendalam memungkinkan pembelajaran otomatis ini dengan menyerap sejumlah besar data tidak terstruktur seperti teks, gambar, dan video [23].

2.3 Python

Python dikembangkan oleh Guido van Rossum di Laboratorium Nasional Belanda pada akhir 1980-an. Versi pertama secara resmi dirilis pada tahun 1991. *Syntax* inti diturunkan dari bahasa ABC dan banyak meminjam dari C dan bahasa lain seperti C++, Modula-3, Unix Shell, SmallTalk, dan Algol-68. *Python* adalah bahasa pemrograman tujuan umum, berorientasi objek dan interaktif.

```
print('PROGRAM MENGHITUNG BMI ')
print("cat : gunakan titik(.) sebagai pengganti koma(,) \n")

bb = float(input('Masukan Berat Badan (kg) : \n'))
tb = float(input('Masukan Tinggi Badan (cm): \n'))

bmi = bb/(tb/100)**2 #Body Mass Index , Rumus BMI jika tinggi
print('nilai BMI kamu adalah', bmi, '\n')

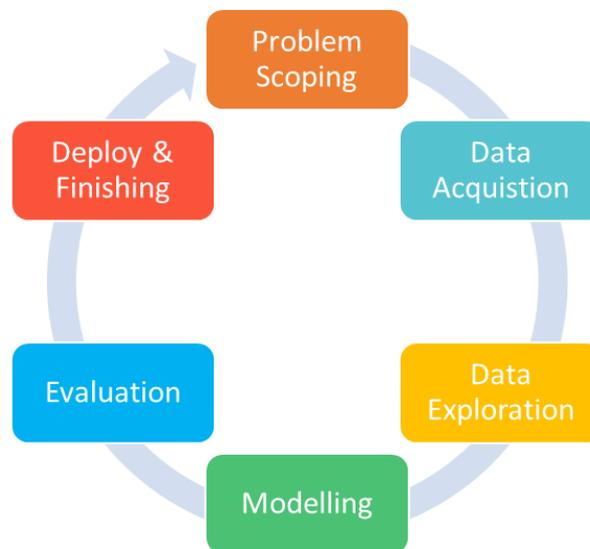
if bmi > 30 :
| print('Status BMI : Obesitas Tingkat 2')
elif bmi >25:
| print('Status BMI : Obesitas Tingkat 1 ')
elif bmi >23:
| print('status BMI : Kelebihan Berat Badan')
elif bmi >18.5:
| print('Status BMI : Normal')
else :
| print('Status BMi : Kurang Berat Badan')
```

Gambar 2. 3 Contoh Kode *Python*

Gambar 2.3 adalah contoh kode *Python*, dapat terlihat perintah perintah dasar *Python* seperti *print*, *if*, dan *elif* serta perhitungan matematika. Kode *Python* adalah bahasa skrip tingkat tinggi dan dirancang agar mudah dibaca. Program *Python* sebagian besar ditulis dalam bahasa Inggris standar dan memiliki indentasi. *Python* menggunakan kata kunci bahasa Inggris, kode ringkas, dan *syntax* sederhana. *Python* adalah bahasa pemrograman multi-paradigma yang mendukung teknik pemrograman terstruktur, fungsional, berorientasi aspek, dan berorientasi objek [2].

Salah satu alasan utama popularitas *Python* adalah perpustakaan standarnya yang luas dengan lebih dari 100 modul. Pengguna dapat dengan mudah mengakses modul ini dengan kode impor sederhana. Yang paling umum digunakan termasuk pemrograman jaringan, protokol Internet seperti HTTP, SMTP, FTP, fungsi matematika, fungsi acak, pencocokan ekspresi reguler, sistem operasi, *toolkit* GUI, *parsing* HTML, pemrosesan XML, email, dll.

2.4 *AI Project Life Cycle*



Gambar 2. 4 Alur Kerja *AI Project Life Cycle* [1]

Gambar 2.4 menunjukkan alur *AI Project Life Cycle* atau Siklus hidup proyek AI, *AI Project Cycle* adalah fase atau metode kerja untuk membuat proyek AI. Siklus hidup proyek AI dapat dibagi menjadi lima fase seperti yang ditunjukkan pada diagram di bawah ini [1] :

1. *Problem Scoping*

Pada fase ini, diidentifikasi masalah yang akan diselesaikan dengan bantuan AI. Pada tahap ini proses yang dilakukan adalah mendefinisikan 4W (*What, Who, Why & Where*) yaitu apa masalah yang dibahas, siapa yang mengalami

masalah tersebut, mengapa masalah tersebut ada, dan di mana masalah tersebut dapat terjadi.

2. *Data Acquisition*

Selama fase ini, proses pengumpulan data dilakukan. Tujuan dari fase ini adalah untuk menangkap data untuk proyek yang sedang dibangun. Data dapat berupa informasi atau fakta dan statistik yang dikumpulkan untuk tujuan referensi atau analisis.. Data yang diperlukan untuk melatih model untuk proyek AI yang efisien harus terkait dengan masalah yang diidentifikasi sebelumnya. Data bisa berasal dari berbagai sumber seperti *website*, sensor, survei, *web scraping*, kamera, API, dll.

3. *Data Exploration*

Fase ini bertujuan untuk memvisualisasikan data dari data yang terkumpul untuk mengidentifikasi tren dan pola dalam data. Pada fase ini juga dilakukan *preprocessing data*.

4. *Modelling*

Pada fase ini yang dilakukan adalah pencarian metode dan algoritma AI terbaik untuk mengolah data dan menangani masalah, pada tahap ini juga dilakukan *training* dan pengkodean.

5. *Evaluation*

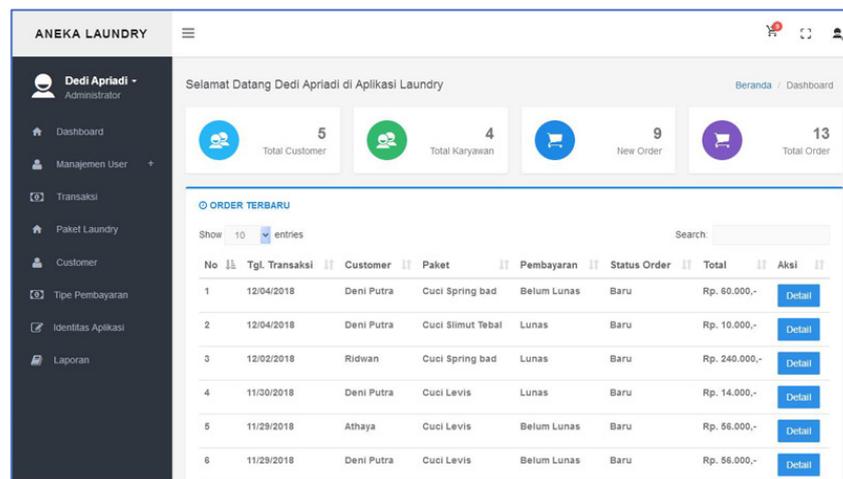
Fase ini adalah tentang memeriksa kinerja model yang dilatih menggunakan data pelatihan. Hal yang paling umum dilakukan pada fase ini adalah perhitungan akurasi, hal ini dilakukan untuk mengukur kinerja model AI yang dibuat untuk menuju ke tahap *deployment*.

6. *Deployment & Finishing*

Pada fase ini dilakukan deployment model yang telah dibuat pada suatu *platform*, baik itu web maupun berupa aplikasi.

2.5 Aplikasi Web

Pengembangan web adalah pekerjaan yang terlibat dalam mengembangkan situs Web untuk Internet (*World Wide Web*) atau intranet (jaringan pribadi). Pengembangan web dapat mengacu dari mengembangkan satu halaman statis sederhana dari teks biasa hingga aplikasi Internet berbasis web yang kompleks (aplikasi web), bisnis elektronik, dan layanan jejaring sosial. Daftar tugas yang lebih komprehensif berkaitan pengembangan eb yaitu dapat mencakup rekayasa web, desain web, pengembangan konten web, penghubung klien, skrip sisi-klien / sisi-server, konfigurasi keamanan server dan jaringan web, dan pengembangan e-niaga [4] .



No	Tgl. Transaksi	Customer	Paket	Pembayaran	Status Order	Total	Aksi
1	12/04/2018	Deni Putra	Cuci Spring bad	Belum Lunas	Baru	Rp. 60.000,-	Detail
2	12/04/2018	Deni Putra	Cuci Slimut Tebal	Lunas	Baru	Rp. 10.000,-	Detail
3	12/02/2018	Ridwan	Cuci Spring bad	Lunas	Baru	Rp. 240.000,-	Detail
4	11/30/2018	Deni Putra	Cuci Levis	Lunas	Baru	Rp. 14.000,-	Detail
5	11/29/2018	Athaya	Cuci Levis	Belum Lunas	Baru	Rp. 56.000,-	Detail
6	11/29/2018	Deni Putra	Cuci Levis	Belum Lunas	Baru	Rp. 56.000,-	Detail

Gambar 2. 5 Contoh Aplikasi Web.

Gambar 2.5 adalah contoh tampilan halaman sebuah aplikasi web. Di antara profesional web, "pengembangan web" biasanya mengacu pada aspek non-desain utama dalam membangun situs web: menulis markup dan coding. Untuk organisasi dan bisnis yang lebih besar, tim pengembangan web dapat terdiri dari ratusan orang (pengembang web) dan mengikuti metode standar seperti metodologi *Agile* saat mengembangkan situs web. Organisasi yang lebih kecil biasanya hanya memerlukan satu pengembang permanen atau kontrak, atau penugasan sekunder untuk posisi pekerjaan terkait seperti desainer grafis atau teknisi sistem informasi.

Pengembangan web juga dapat merujuk pada upaya kolaboratif antar departemen daripada domain departemen yang ditunjuk. Ada tiga jenis spesialisasi pengembang web: pengembang *front-end*, pengembang *back-end*, dan pengembang *full-stack*. Pengembang *front-end* bertanggung jawab atas perilaku dan visual yang berjalan di browser pengguna, sedangkan pengembang *back-end* menangani server.

2.6 Flask

Flask adalah kerangka kerja mikro pengembangan web yang ditulis dengan *Python*. Mudah dipelajari dan digunakan. *Flask* "ramah pemula" karena tidak ada kode *boilerplate* atau dependensi yang mengalihkan perhatian dari fungsionalitas inti aplikasi, Kerangka kerja mikro berbeda dari kerangka kerja *full-stack* yang juga menyediakan modul tambahan untuk fitur seperti otentikasi, ORM basis data, validasi input, sanitasi, dll.

```

from flask import Flask, render_template, request
import jsonify
import requests
import pickle
import numpy as np

app = Flask(__name__)
model = pickle.load(open('color.pkl', 'rb'))

@app.route('/', methods=['GET'])
def Home():
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'GET':
        message = {'answer': 'Your answer is showed here'}
        return message
    if request.method == 'POST':
        data = str(request.data)
        # answer = model.predict([[request.body]]);
        predictionData = data[3:-2].split(',')
        r = int(predictionData[0])
        g = int(predictionData[1])
        b = int(predictionData[2])
        answer = model.predict([[r, g, b]])
        # print(answer)
        message = {'answer': answer[0]}
        return message

if __name__ == "__main__":
    app.run(debug=True)

```

Gambar 2. 6 Contoh Penerapan *Flask*

Gambar 2.6 adalah contoh *microframework Flask*, dapat dilihat bagian utama *Flask* yaitu “*app = Flask(__name__)*” yang menyatakan pembuatan aplikasi *Flask*. *Flask* lahir pada tahun 2004 ketika seorang pengembang bernama Armin Ronacher membuat lelucon *April Mop*. Namun, dengan cepat mendapatkan popularitas di komunitas *open source*. Akibatnya, *Flask* tumbuh menjadi proyek *open source* populer yang berlanjut hingga hari ini.

Flask dikenal sebagai *micro-framework* karena ringan dan hanya berisi komponen yang diperlukan. *Flask* hanya menyediakan komponen yang dibutuhkan untuk pengembangan web, seperti: *routing*, pemrosesan *request*, sesi, dll. Untuk fungsi lain seperti pemrosesan data, pengembang dapat menulis modul mereka sendiri atau menggunakan ekstensi. Pendekatan ini menghindari kode *boilerplate* yang tidak perlu yang sebenarnya tidak digunakan [3].

Berikut adalah beberapa fitur yang menjadikan *Flask* sebagai *framework* ideal untuk pengembangan aplikasi web.

1. *Flask* menyediakan server pengembangan dan *debugger*.
2. Menggunakan template Jinja2.
3. Mendukung WSGI 1.0.
3. Dukungan bawaan untuk pengujian unit.
4. *Flask* menyediakan banyak ekstensi yang dapat digunakan untuk memperluas fungsionalitasnya.

2.7 *HTML*

Hypertext Markup Language (*HTML*) adalah bahasa *markup* standar untuk dokumen yang dirancang untuk ditampilkan di *browser web*. *HTML* dapat dibantu oleh teknologi seperti *Cascading Style Sheets* (*CSS*) dan bahasa skrip seperti *JavaScript*.

```

<section class="member">

  <h1 class="heading-title"> Our Members </h1>

  <div class="box-container">
    <div class="box">
      
      <h3>Ezra Taufiqurrahman <br> (Enigma)</h3>
    </div>

    <div class="box">
      
      <h3>Kadafi Eka Sakti <br> (Enigma)</h3>
    </div>

    <div class="box">
      
      <h3>Andjani Ayu Cahaya Tanjung<br>(Enigma)</h3>
    </div>

    <div class="box">
      
      <h3>Diana Rahmawati<br>(Artemis)</h3>
    </div>

    <div class="box">
      
      <h3>Daffa Al Hafizh<br>(Enigma)</h3>
    </div>
  </div>

```

Gambar 2. 7 Contoh *Script* HTML.

Gambar 2.7 adalah contoh potongan kode HTML dapat dilihat *tag* umum HTML seperti *div*, *section* dan *h1*. *Browser web* menerima dokumen HTML dari *server web* atau dari penyimpanan lokal dan membuat dokumen tersebut menjadi halaman web multimedia. HTML mendeskripsikan struktur halaman web secara semantik dan awalnya menyertakan isyarat untuk tampilan dokumen [4].

Elemen HTML adalah blok pembangun halaman HTML. Dengan konstruksi HTML, gambar dan objek lain seperti bentuk interaktif dapat disematkan

ke halaman yang dirender. HTML menyediakan sarana untuk membuat dokumen terstruktur dengan menunjukkan semantik struktural untuk teks seperti judul, paragraf, daftar, tautan, kutipan, dan item lainnya. Elemen HTML digambarkan dengan *tag*, ditulis menggunakan tanda kurung sudut. *Tag* seperti `` dan `<input />` secara langsung memasukkan konten ke dalam halaman. Tag lain seperti `<p>` mengelilingi dan memberikan informasi tentang teks dokumen dan mungkin menyertakan *tag* lain sebagai sub-elemen. Browser tidak menampilkan tag HTML, tetapi menggunakannya untuk menafsirkan konten halaman.

HTML dapat menyematkan program yang ditulis dalam bahasa skrip seperti *JavaScript*, yang memengaruhi perilaku dan konten halaman web. Penyertaan CSS digunakan untuk menentukan tampilan dan tata letak konten. *World Wide Web Consortium* (W3C), mantan pengelola HTML dan pengelola standar CSS saat ini, telah mendorong penggunaan CSS di atas HTML sejak 1997.

2.8 CSS

Cascading Style Sheets (CSS) adalah bahasa *style sheet* yang digunakan untuk menggambarkan presentasi / tampilan dokumen yang ditulis dalam bahasa markup seperti HTML. CSS adalah teknologi landasan dari *World Wide Web*, bersama dengan HTML dan *JavaScript*.

```
.colorBox {  
  width: 100px;  
  height: 1.25rem;  
  display: inline-flex;  
  margin-left: 5px;  
  border: 2px solid ■ black;  
}  
  
.answer {  
  display: flex;  
  justify-content: center;  
}  
  
.image {  
  max-width: 100%;  
  height: auto;  
}  
  
html {  
  height: 100%;  
}
```

Gambar 2. 8 Contoh Script CSS..

Gambar 2.8 adalah contoh dokumen CSS yang mengatur tampilan dokumen HTML menggunakan *selector* CSS seperti `.`, `#` dan `dll`. CSS dirancang untuk memungkinkan pemisahan presentasi dan konten, termasuk tata letak, warna, dan font. Pemisahan ini dapat meningkatkan aksesibilitas konten, memberikan lebih banyak fleksibilitas dan kontrol dalam spesifikasi karakteristik presentasi, memungkinkan beberapa halaman web untuk berbagi pemformatan dengan menentukan CSS yang relevan dalam file `.css` terpisah yang mengurangi kompleksitas dan pengulangan dalam konten struktural serta memungkinkan file `.css` yang akan disimpan dalam cache untuk meningkatkan kecepatan memuat halaman antara halaman [4] .

2.9 JavaScript

JavaScript atau yang sering disingkat JS, adalah bahasa pemrograman yang sesuai dengan spesifikasi ECMAScript. *JavaScript* adalah bahasa pemrograman tingkat tinggi dan multi-paradigma. JS memiliki *syntax* kurung kurawal, pengetikan dinamis, orientasi objek berbasis prototipe, dan fungsi kelas satu [4].

Selain HTML dan CSS, *JavaScript* adalah salah satu teknologi inti dari *World Wide Web*. *JavaScript* memungkinkan halaman web yang interaktif dan merupakan bagian penting dari aplikasi web. Sebagian besar situs web menggunakannya untuk perilaku halaman sisi klien, dan semua *browser* web utama memiliki mesin *JavaScript* khusus untuk menjalankannya.

```
function loadImage(src) {
  introduction.hidden = true;
  image.hidden = false;
  image.src = src;
}

function loadFile(e) {
  if (e.target.files) {
    let imageFile = e.target.files[0];
    let reader = new FileReader();
    reader.readAsDataURL(imageFile);
    reader.onloadend = function (e) {
      loadImage(e.target.result);
    }
  }
}

function getColor(x, y) {
  let pixel = c.getImageData(x, y, 1, 1);
  let rgb = pixel.data;

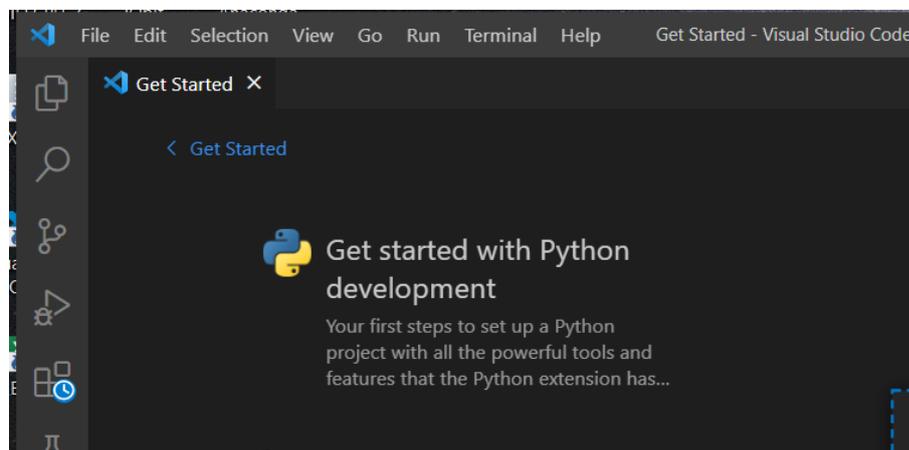
  return rgb;
}
```

Gambar 2. 9 Contoh Script JavaScript.

Gambar 2.9 adalah contoh kode bahasa pemrograman *JavaScript*, pada contoh diatas dapat dilihat bentuk fungsi dalam *JavaScript*. Sebagai bahasa multi-paradigma, *JavaScript* mendukung gaya pemrograman yang digerakkan oleh peristiwa, fungsional, dan imperatif. JS memiliki antarmuka pemrograman aplikasi (API) untuk bekerja dengan teks, tanggal, ekspresi reguler, struktur data standar, dan Model Objek Dokumen (DOM). Namun, bahasanya sendiri tidak menyertakan *input / output* (I / O) apa pun, seperti jaringan, penyimpanan, atau fasilitas grafik, karena *environment host* (biasanya browser web) menyediakan API tersebut .

2.10 *Visual Studio Code*

Visual Studio Code adalah editor *source-code* gratis yang dibuat oleh *Microsoft* untuk *Windows*, *Linux*, dan *macOS*. Fitur-fiturnya termasuk dukungan untuk *debugging*, *syntax highlighting*, *intelligent code complete*, *snippet*, *code refactoring*, dan *embedded Git*. Pengguna dapat mengubah tema, pintasan *keyboard*, preferensi, dan memasang ekstensi yang menambahkan fungsionalitas tambahan [6].



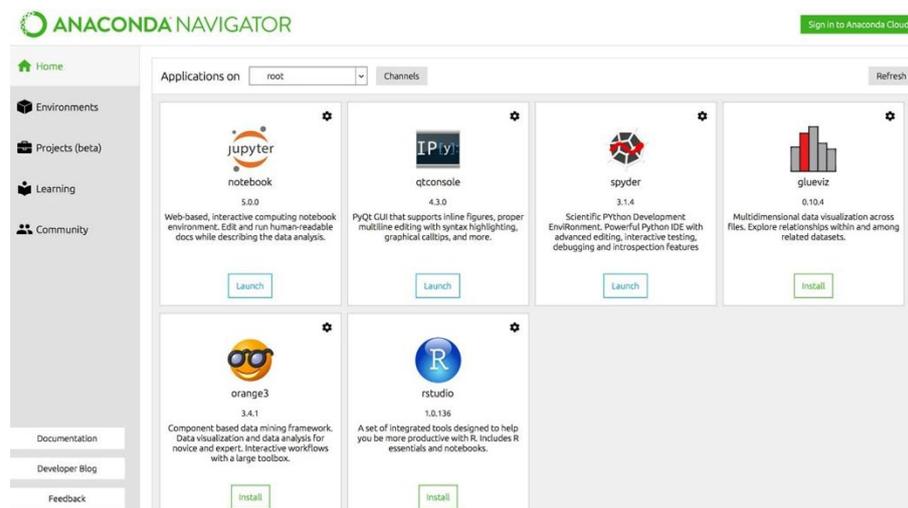
Gambar 2. 10 Antarmuka IDE *Visual Studio Code*.

Gambar 2.10 adalah tampilan halaman utama *Visual Studio Code* pada komputer. *Visual Studio Code* dapat digunakan dengan berbagai bahasa pemrograman, termasuk *Java*, *JavaScript*, *Go*, *Node.js* dan *C ++*. Aplikasi ini dibuat berdasarkan *framework Electron*, yang digunakan untuk mengembangkan

aplikasi *Web Node.js* yang berjalan pada *Blink*. *Visual Studio Code* menggunakan komponen editor yang sama (dengan nama kode "Monaco") yang digunakan di *Azure DevOps*.

2.11 Anaconda

Anaconda adalah aplikasi distribusi *open-source* dari bahasa pemrograman *Python* dan *R* untuk ilmu data yang bertujuan menyederhanakan manajemen dan penyebaran paket. Versi paket pada *Anaconda* dikelola oleh sistem manajemen paket *Conda*. Sistem menganalisis *environment* saat ini sebelum penginstalan untuk menghindari kerusakan kerangka kerja dan paket lain [7].

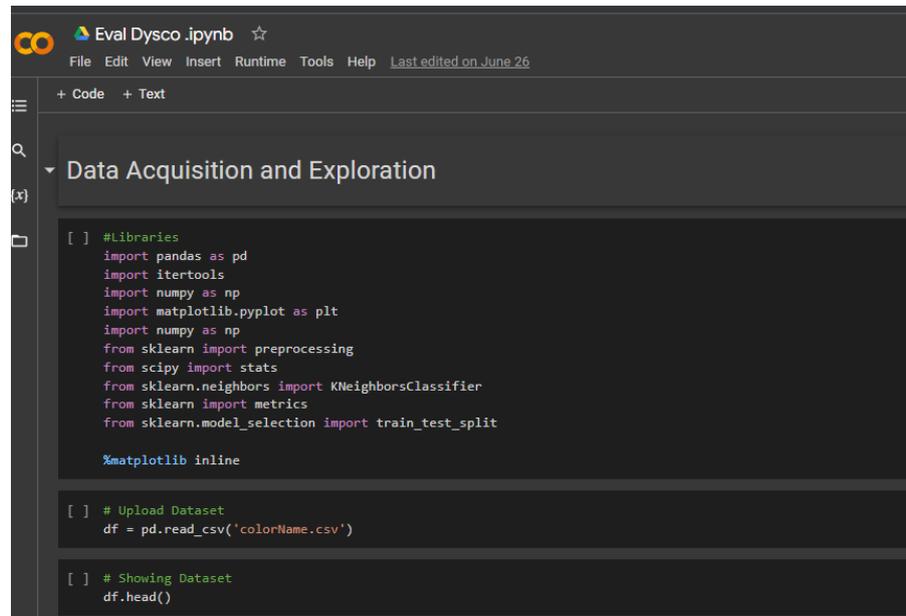


Gambar 2. 11 Tampilan UI Anaconda Navigator.

Gambar 2.11 adalah tampilan halaman utama *Anaconda* pada komputer. Lebih dari 250 paket diinstal secara otomatis di distribusi *Anaconda*. Selain paket *Conda* dan *Virtual Environment Manager*, lebih dari 7500 paket *open source* tambahan dapat diinstal dari PyPI. *Anaconda* juga termasuk *Anaconda Navigator*, GUI (*Graphical User Interface*), sebagai alternatif grafis untuk antarmuka baris perintah. *Anaconda Navigator* termasuk dalam distribusi *Anaconda* dan memungkinkan pengguna untuk meluncurkan aplikasi dan mengelola paket *Conda*, *environment*, dan saluran tanpa menggunakan perintah baris perintah. *Navigator*

dapat mencari paket, menginstalnya di *environment* , menjalankan paket, dan memperbaruinya.

2.12 Google Collab

The image shows a screenshot of the Google Colab web interface. At the top, the title bar reads 'Eval Dysco .ipynb' with a star icon and a 'Last edited on June 26' timestamp. Below the title bar is a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The main area is a Jupyter Notebook with a dark theme. The notebook title is 'Data Acquisition and Exploration'. The code in the notebook is as follows:

```
[ ] #Libraries
import pandas as pd
import itertools
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
from sklearn import preprocessing
from scipy import stats
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split

%matplotlib inline

[ ] # Upload Dataset
df = pd.read_csv('colorName.csv')

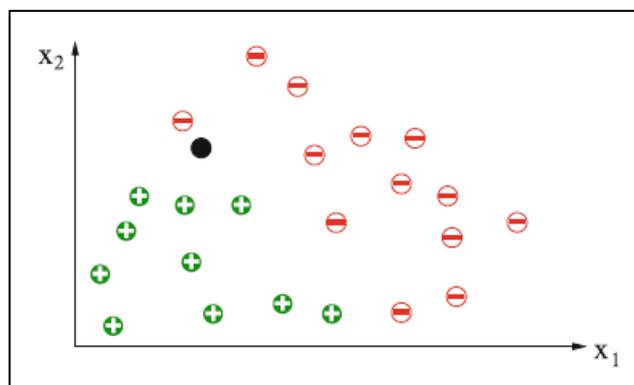
[ ] # Showing Dataset
df.head()
```

Gambar 2. 12 Tampilan UI *Google Collab*.

Gambar 2.12 adalah tampilan dari *website Google Collab* pada *web browser*. *Google Colab* dikembangkan oleh *Google* untuk memberikan akses gratis ke GPU dan TPU bagi siapa saja yang membutuhkannya untuk membuat model pembelajaran mesin atau pembelajaran mendalam. *Google Colab* dapat didefinisikan sebagai versi yang disempurnakan dari *Jupyter Notebook*. *Jupyter Notebook* adalah aplikasi yang memungkinkan untuk mengedit dan menjalankan dokumen *Notebook* melalui *browser web* atau *environment* pengembangan terintegrasi (IDE). Bekerja di buku catatan, bukan file [8].

2.13 *K-Nearest Neighbor*

KNN (*K-Nearest Neighbor*) adalah algoritma pembelajaran AI, algoritma ini termasuk dalam *supervised learning*. Algoritma ini mengasumsikan bahwa tiap *instance* dari data adalah bertetangga dan *instance* yang berdekatan dianggap memiliki kelas yang sama, tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel dari *training* data [10].



Gambar 2. 13 Ilustrasi Klasifikasi K-NN [23]

Gambar 2.13 adalah ilustrasi klasifikasi menggunakan K-NN dalam iluastasi digunakan dua kelas yang disimbolkan dengan + dan - .Algoritma KNN menggunakan *Neighborhood Classification* sebagai nilai prediksi dari nilai *isntance* yang baru. Secara umum alur kerja K-NN adalah sebagai berikut :

1. Menentukan jumlah tetangga (K) yang akan digunakan untuk pertimbangan penentuan kelas
2. Menghitung jarak dari data baru ke masing masing data pada *dataset* untuk menentukan tetangga terdekat
3. Menentukan nilai jarak terdekat dan penentuan kelas.
4. Klasisfikasi dengan mengambil nilai K, kemudian menentukan kelas dari data baru

Rumus yang digunakan untuk menentukan jarak adalah dengan perhitungan *Euclidean Distance* seperti berikut :

$$dis(x_1, x_2) = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2}$$

Gambar 2. 14 Rumus *Euclidean Distance* 1 Variabel [23]

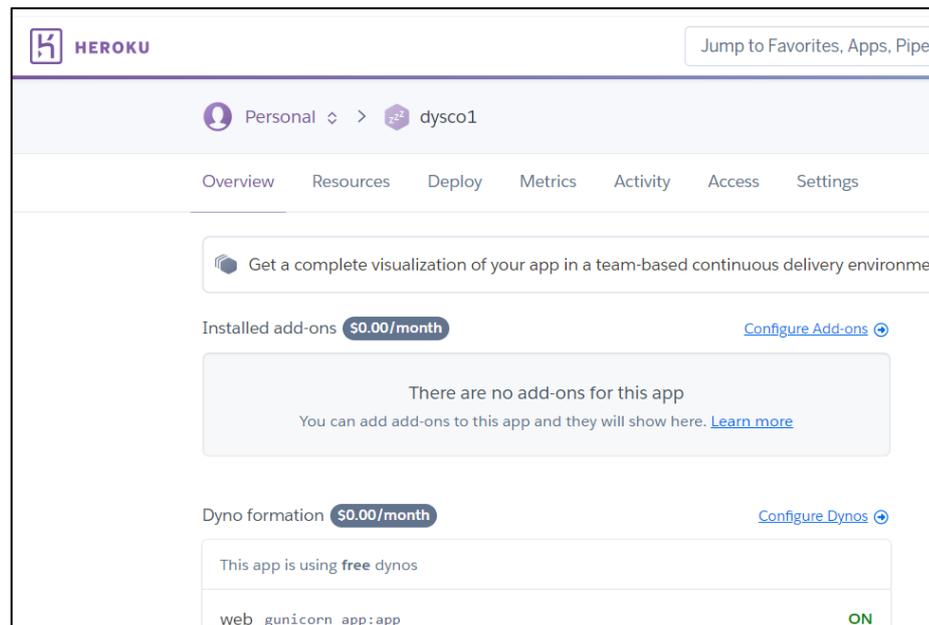
$$dis = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2 + \dots}$$

Gambar 2. 15 Rumus *Euclidean Distance* n variabel [23]

Pada gambar 2.14 dan 2.15 diatas x_1 adalah data yang akan di prediksi dan adalah x_2 data pada *entry* pertama. Dalam prakteknya di python dapat digunakan kode program untuk mendapatkan nilai K terbaik dari berbagai opsi [23].

2.14 HEROKU

Heroku (<http://www.heroku.com>), diucapkan her-oh-koo, adalah salah satu penyedia PaaS terkemuka dalam bisnis perangkat lunak *cloud* dan merupakan solusi PaaS terkemuka untuk bisnis kecil dan perusahaan besar. Dengan peningkatan konstan dan filosofi 'kenyamanan' daripada 'konfigurasi', Heroku telah menjadi platform pengembangan aplikasi *cloud* terkemuka untuk pengembang dan digunakan oleh lebih dari 40.000 situs webnya hingga saat ini. Filosofi Heroku adalah memungkinkan pengembang untuk melupakan server dan hanya fokus pada pembuatan aplikasi webnya. Heroku secara ajaib menangani pembuatan, penerapan, pengoperasian, dan penskalaan aplikasi sesuai kebutuhan.



Gambar 2. 16 Tampilan UI HEROKU

Gambar 2.16 adalah tampilan halaman utama website Heroku pada web browser. Heroku adalah *platform* aplikasi *cloud* multibahasa yang memberi fleksibilitas tinggi dalam memilih bahasa pemrograman yang tepat untuk pengembangan aplikasi web. Sejak awal 2013, Heroku telah menyediakan dukungan platform untuk *Ruby*, *Ruby on Rails*, *Java*, *Node.js*, *Clojure*, *Scala*, *Python*, dan *PHP* [14].

2.15 Usability Testing

Usability Testing merupakan proses belajar tentang pengguna dari pengguna dengan mengamati mereka menggunakan produk untuk mencapai tujuan tertentu .

Menyiapkan *Usability Testing* melibatkan pembuatan skenario dengan hati-hati, atau situasi realistis, di mana orang tersebut melakukan daftar tugas menggunakan produk yang sedang diuji sementara pengamat menonton dan membuat catatan (verifikasi dinamis). Beberapa instrumen tes lain seperti instruksi tertulis, prototipe kertas, dan kuesioner pra dan pasca tes juga digunakan untuk

mengumpulkan *feedback* pada produk yang diuji (verifikasi statis). Tujuannya adalah untuk mengamati bagaimana pengguna berfungsi secara realistis, sehingga pengembang dapat mengidentifikasi area masalah dan memperbaikinya. Teknik yang populer digunakan untuk mengumpulkan data selama *Usability Testing* termasuk protokol *critical thinking* dan *questionare*.

Menurut ISO 9241-11 [24], ada tiga aspek yang dijadikan acuan untuk mengukur seberapa bergunanya aplikasi dalam membantu mencapai tujuan tertentu oleh pengguna aplikasi. Ketiga aspek usability tersebut, yaitu: *Effectiveness* (kemampuan pengguna untuk mencapai tujuan mereka dalam konteks yang spesifik), *Efficiency* (sumber daya yang digunakan terkait dengan akurasi dan kesempurnaan yang dicapai pengguna dalam menjalankan tugas), dan *Satisfaction* (tingkat kepuasan pengguna saat menggunakan aplikasi). *Usability Testing* adalah pengujian penggunaan terhadap sistem atau produk untuk menemukan permasalahan daya guna atau *usability*. Dalam pengujian *usability* terdapat satu teknik yang bisa digunakan dalam mengukur tingkat efektifitas dan efisiensi suatu produk, yaitu: teknik *performance measurement*. Teknik tersebut dapat memberikan hasil pengukuran yang lebih akurat dikarenakan sistem pengujiannya langsung berdasarkan pengalaman penggunanya. Di sisi lain, *System Usability Scale* (SUS) dapat memberikan pandangan subjektif tentang usability dari suatu sistem [25]. Nilai yang didapatkan dari metode SUS dapat dijadikan pertimbangan tingkat kelayakan suatu aplikasi untuk diterapkan. Oleh karena itu, kuesioner SUS dapat digunakan untuk mengukur tingkat kepuasan pengguna terhadap suatu produk karena sifat penilaiannya yang subjektif.

1. Teknik *Performance Measurement*

Untuk melihat keefektifan akan dilihat dari jumlah *task* yang berhasil diselesaikan responden. Penyelesaian dihitung dengan menetapkan angka biner '1' jika partisipan berhasil dan '0' jika partisipan gagal. Rumus yang digunakan adalah sebagai berikut:

$$\text{Efektifitas} = \frac{\text{Jumlah tugas yang berhasil diselesaikan}}{\text{Jumlah total tugas}} \times 100\%$$

Gambar 2. 17 Rumus Penilaian Efektifitas [29]

Pada gambar 2.17 diatas dijabarkan untuk memperoleh hasil efektifitas, banyaknya tugas yang diselesaikan dan berhasil dibagi banyaknya total tugas yang diberikan yang hasilnya dikalikan 100%. Rata-rata penyelesaian tugas minimum pada pengujian *usability* adalah 78%, namun bila hasil dibawah 49% menempatkan pada kuartil bawah [29].

Efektivitas harus dinilai atas dasar tujuan yang bisa dilaksanakan, bukan atas dasar konsep tujuan yang maksimum. Efektivitas diukur dengan menggunakan standar sesuai dengan acuan Litbang Depdagri (1991) seperti pada Tabel 2.1 [27]

Tabel 2. 1 Standar Ukuran Efektifitas [27]

Rasio Efektivitas	Tingkat Capaian
Dibawah 40	Sangat tidak efektif
40 – 59.99	Tidak Efektif
60 – 79.99	Cukup Efektif
Diatas 80	Sangat Efektif

Sedangkan untuk mengukur Keefisiensian dapat diukur dalam waktu tugas, yaitu waktu dalam hitungan menit/detik yang dibutuhkan untuk menyelesaikan tugas dengan sukses. Rumus yang digunakan untuk menghitung keefisienan suatu produk atau layanan adalah sebagai berikut: Waktu yang diperoleh diukur dengan mengurangi waktu selesai dengan waktu mulai, kemudian dihitung menggunakan persamaan pada gambar 2.18 berikut.

$$\text{Efisiensi efektif keseluruhan} = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR}$$

Gambar 2. 18 Rumus Perhitungan Efisiensi [29]

Keterangan :

N = Jumlah total tugas (gol)

R = Jumlah pengguna

n_{ij} = Hasil tugas i oleh pengguna j ; jika pengguna berhasil menyelesaikan tugas, maka $N_{ij} = 1$, jika tidak, maka $N_{ij} = 0$, t_{ij} = Waktu yang dihabiskan oleh pengguna j untuk menyelesaikan tugas i . Jika tugas tidak berhasil diselesaikan, maka waktu diukur hingga saat pengguna berhenti dari tugas .

Selanjutnya hasil rata-rata waktu yang dibutuhkan untuk mengerjakan *task scenario* tersebut diinterpretasikan menggunakan *range* waktu pada indikator *time behavior* seperti yang terdapat pada Tabel 2. 2 untuk menentukan durasi waktu yang digunakan pengguna dalam penyelesaian task sehingga dapat diketahui hasil pengukuran tingkat efisiensi dari aplikasi [28].

Tabel 2. 2 Standar Ukuran Efisiensi [28]

No	Lamanya Waktu	Kualifikasi
1	60 -300 <i>Second</i>	Sangat Cepat
2	360 - 600 <i>Second</i>	Cepat
3	660 - 900 <i>Second</i>	Lambat

2. System Usability Scale

Salah satunya pengujian *usability* bisa dilakukan menggunakan *System Usability Scale* (SUS) [18]. SUS memiliki 10 pertanyaan dan 5 pilihan jawaban. Pilihan jawaban terdiri dari sangat tidak setuju sampai sangat setuju. SUS memiliki skor minimal 0 dan skor maksimal 100. SUS dalam bahasa aslinya menggunakan

bahasa Inggris. Berikut 10 pertanyaan dari *System Usability Scale* (SUS) yang sudah diterjemahkan dalam bahasa Indonesia [19]:

No.	Item in Indonesian
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini rumit untuk digunakan.
3	Saya merasa sistem ini mudah untuk digunakan.
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Gambar 2. 19 Pertanyaan SUS [19]

Gambar 2.19 diatas menampilkan daftar 10 pertanyaan SUS. SUS memiliki 5 pilihan jawaban. Mulai dari sangat tidak setuju, tidak setuju, ragu-ragu, setuju, dan sangat setuju. Skor masing-masing jawaban mulai dari 1 sampai 5. Berikut pilihan jawaban beserta skornya.

Jawaban	Skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-ragu (RG)	3
Setuju (S)	4
Sangat Setuju (SS)	5

	STS	TS	RG	ST	SS
1. Saya berpikir akan menggunakan sistem ini lagi.	<input type="text"/>				
	1	2	3	4	5

Gambar 2. 20 Chart Nilai Pertanyaan SUS [18]

Gambar 2.20 adalah sebuah *chart* penilaian setiap jawaban SUS. Setelah melakukan pengumpulan data dari responden, kemudian data tersebut dihitung. Dalam cara menggunakan *System Usability Scale* (SUS) ada beberapa aturan dalam perhitungan skor SUS. Berikut ini aturan-aturan saat perhitungan skor pada kuesionernya:

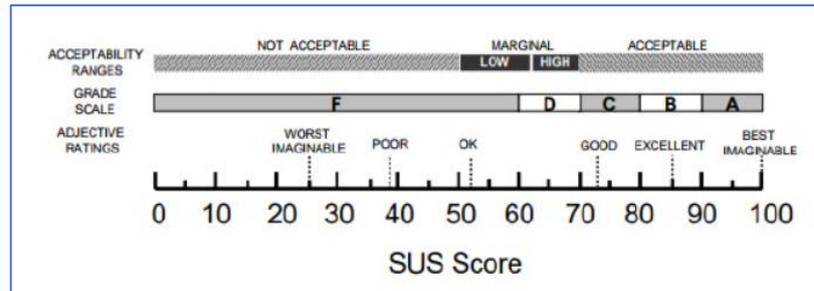
- a. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1.
- b. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
- c. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

$$\bar{x} = \frac{\sum x}{n}$$

\bar{x}	=	skor rata-rata
$\sum x$	=	jumlah skor SUS
n	=	jumlah responden

Gambar 2. 21 Rumus Perhitungan SUS [18]

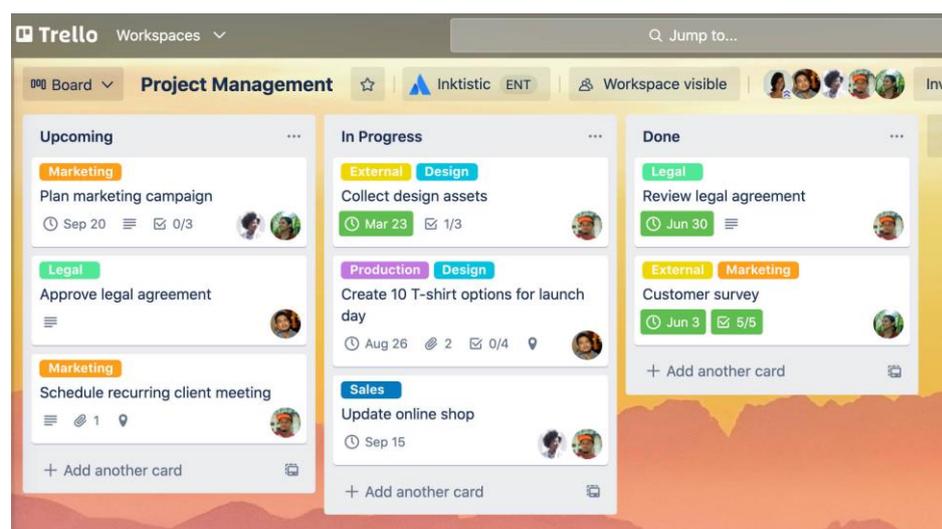
Gambar 2.21 menunjukkan rumus perhitungan SUS yaitu adalah rata rata dari seluruh jumlah skor sus dibagi dengan jumlah responden. Aturan perhitungan skor untuk berlaku pada 1 responden. Untuk perhitungan selanjutnya, skor SUS dari masing-masing responden dicari skor rata-ratanya dengan menjumlahkan semua skor dan dibagi dengan jumlah responden.



Gambar 2. 22 Grafik Kesimpulan SUS [18]

Gambar 2.22 adalah grafik kesimpulan nilai skor SUS ada 3 teknik penilaian yaitu berdasarkan *acceptability range*, *grade scale* dan *adjective ratings*. Kesimpulan dari cara menggunakan *System Usability Scale* (SUS) adalah setelah dihitung didapatlah skor rata-rata SUS dari semua responden. Skor tersebut kemudian disesuaikan dengan penilaian SUS. Masuk kategori mana hasil pengujian dengan skor rata-rata yang sudah didapat. Skor rata-rata SUS dari banyaknya penelitian adalah 68, maka jika nilai SUS di atas 68 akan dianggap di atas rata-rata dan nilai di bawah 68 di bawah rata-rata. Jika skor yang kamu dapat dibawah 68 berarti ada masalah pada *usability* dan butuh perbaikan. Namun kesimpulan akhir bisa juga ditentukan melalui penilaian seperti pada gambar diatas [19].

2.16 Trello

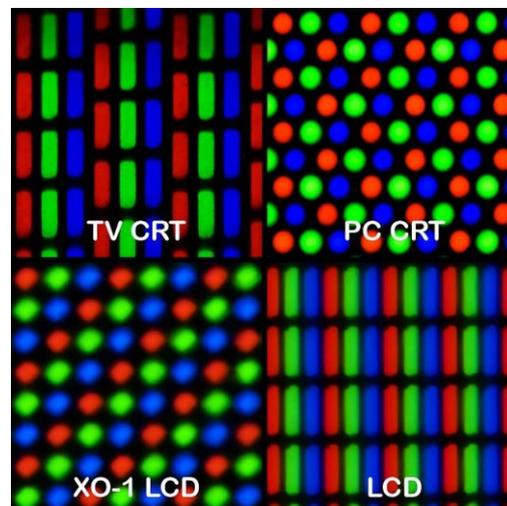


Gambar 2. 23 Tampilan Board Trello

Gamabr 2.23 diatas adalah tampilan utama webiste trello pada *web browser*. *Trello* sebuah alat yang digunakan sebagai papan proyek daring. Trello menyediakan cara mudah, gratis, fleksibel, dan visual untuk mengelola proyek dan mengatur apa pun didalamnya. Manfaat dari penggunaan *trello* adalah dapat digunakan sebagai aplikasi kolaborasi yang memungkinkan semua anggota proyek tersebut bisa tahu apa yang sedang dikerjakan, siapa yang mengerjakannya, kapan pekerjaan tersebut direncanakan selesai, dan sejauh mana orang tersebut mengerjakannya [5].

2.17 Piksel

Dalam pencitraan digital, piksel (disingkat px), *pel*, atau elemen gambar, adalah elemen terkecil yang dapat dialamatkan dalam gambar, atau titik terkecil di perangkat tampilan tempat semua titik dapat dialamatkan. Di sebagian besar perangkat tampilan digital, piksel adalah elemen terkecil yang dapat dimanipulasi dalam perangkat lunak seperti yang dapat dilihat pada Gambar 2.24 berikut [31].



Gambar 2. 24 Tampilan Piksel pada Berbagai *Display* [30].

Setiap piksel adalah sampel dari gambar asli. Jumlah sampel yang lebih banyak umumnya memberikan representasi yang lebih akurat dari aslinya. Setiap piksel memiliki intensitas variabel. Dalam sistem pencitraan warna, warna biasanya diwakili oleh tiga atau empat komponen intensitas. Merah, Hijau, Biru atau *Cyan*,

Magenta, Kuning, Hitam. Definisi piksel sangat peka konteks. Misalnya, "piksel tercetak" dapat berada di halaman, atau berupa piksel yang ditransmisikan oleh sinyal elektronik atau diwakili oleh nilai digital, piksel pada perangkat tampilan, atau piksel dalam kamera digital (elemen sensor cahaya).

Semakin banyak piksel yang digunakan untuk merepresentasikan gambar, maka hasilnya akan semakin mendekati gambar aslinya. Jumlah piksel dalam gambar terkadang disebut resolusi, tetapi resolusi memiliki definisi yang lebih spesifik. Jumlah piksel dapat dinyatakan sebagai angka tunggal, seperti kamera digital "3 megapiksel", yang biasanya berukuran 3 juta piksel, atau sebagai pasangan, seperti "layar 640 x 480" dengan 640 piksel di satu sisi ke-sisi lain dan 480 piksel atas ke bawah, total $640 \times 480 = 307.200$ piksel atau 0,3 megapiksel [32].

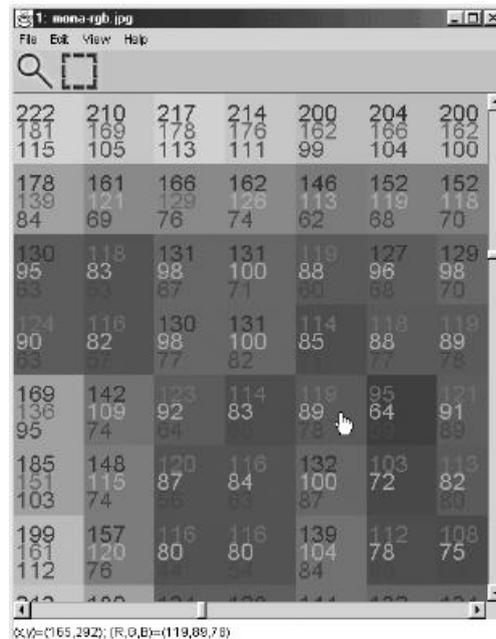
2.18 RGB

Model warna RGB (*Red, Green, Blue*) adalah model warna tambahan di mana warna primer cahaya merah, hijau dan biru ditambahkan bersama-sama dalam berbagai cara untuk mereproduksi susunan warna yang luas. Nama model berasal dari inisial dari tiga warna primer aditif, merah, hijau, dan biru [32].

Tujuan utama dari model warna RGB adalah untuk penginderaan, representasi, dan tampilan gambar dalam sistem elektronik, seperti televisi dan komputer, meskipun juga telah digunakan dalam fotografi konvensional. Sebelum zaman elektronik, model warna RGB sudah memiliki teori yang kuat di belakangnya, berdasarkan persepsi warna manusia.

Perbesar terus sebuah gambar dan akan segera berada pada skala di mana masing-masing piksel akan terlihat. Setiap piksel adalah wilayah persegi kecil dari gambar. Warnanya diwakili oleh tiga angka: komponen merah, komponen hijau, dan komponen biru. ketiga komponen piksel ini dikenal sebagai nilai RGB piksel atau RGB *triple*. Nilai terkecil yang dapat muncul dalam RGB *triple* adalah 0. Misalnya RGB *triple* (0, 0, 0) mewakili warna yang memiliki 0 satuan merah, 0

satuan hijau, dan 0 satuan biru, warna ini adalah hitam., nilai terbesar yang dapat muncul dalam triple RGB adalah 255, dan RGB *triple* (255, 255, 255) mewakili warna putih [33].



Gambar 2. 25 Representasi nilai RGB pada Gambar [33].

Pada Gambar 2.25 menunjukkan nilai setiap piksel dan nilai RGB pada koordinat yang juga merepresentasikan intensitas warna pada citra berwarna yang dapat dikodekan dalam 24 bit, masing masing nilai RGB dikodekan dengan 8 bit, dengan demikian sebuah citra berwarna dapat memiliki kandungan warna maksimum sebanyak 2^{24} atau sebanyak 16.777.216 variasi warna [34].

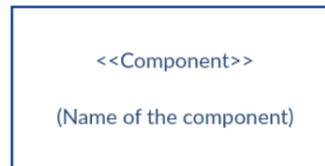
2.20 Diagram Komponen

Diagram komponen digunakan untuk memvisualisasikan organisasi komponen sistem dan hubungan ketergantungan di antara mereka. Diagram ini memberikan pandangan tingkat tinggi dari komponen dalam suatu sistem.

Komponen tersebut dapat berupa komponen perangkat lunak seperti *database* atau antarmuka pengguna; atau komponen perangkat keras seperti sirkuit,

microchip, atau perangkat; atau unit bisnis seperti pemasok, penggajian, atau pengiriman [35].

Berikut adalah beberapa simbol yang digunakan dalam diagram komponen :



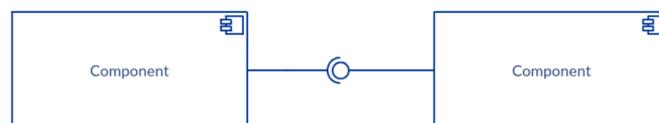
Gambar 2. 26 *Component*

Gambar 2.26 adalah simbol dari *Component* (Komponen). Persegi panjang dengan stereotip komponen (<<component>>). Stereotip komponen biasanya digunakan di atas nama komponen untuk menghindari kebingungan bentuk dengan ikon *class*.



Gambar 2. 27 *Dependencies*

Gambar 2.27 adalah simbol dari *Dependencies* (Kebergantungan). Meskipun dapat menampilkan lebih detail tentang hubungan antara dua komponen menggunakan notasi bola-dan-soket (antarmuka yang disediakan dan antarmuka yang diperlukan), dapat juga digunakan panah *dependencies* untuk menunjukkan hubungan antara dua komponen.



Gambar 2. 28 *Interfaces*

Gambar 2.28 adalah simbol dari *interfaces*. Antarmuka (*interfaces*) dalam diagram komponen menunjukkan bagaimana komponen dihubungkan bersama dan

berinteraksi satu sama lain. Konektor bola dan soket memungkinkan menghubungkan antarmuka yang diperlukan komponen (diwakili dengan setengah lingkaran dan garis padat) dengan antarmuka yang disediakan (diwakili dengan lingkaran dan garis padat) dari komponen lain. Ini menunjukkan bahwa satu komponen menyediakan layanan yang dibutuhkan komponen lainnya.

2.19 Penelitian Terkait

Terdapat beberapa penelitian terkait yang dijadikan sebagai perbandingan serta rujukan mengenai metode serta hasil yang dicapai pada penelitian ini. Berikut merupakan ulasan dari beberapa penelitian terkait:

2.19.1 Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode *K-Nearest Neighbor* (KNN) Berdasarkan Nilai *Hue*, *Saturation*, *Value* (HSV)

Penelitian yang dilakukan oleh Dani Syahid, Jumadi dan Dian Nursantika mengenai Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode *K-Nearest Neighbor* (KNN) Berdasarkan Nilai *Hue*, *Saturation*, *Value* (HSV). Pada penelitian ini sistem digunakan untuk membedakan tiap jenis tanaman hias berdaun *philodendron* berdasarkan nilai HSV atau intensitas warnanya, sistem yang digunakan membandingkan data citra yang telah ada dengan data citra yang baru, untuk mendapatkan nilai HSV sebelumnya dilakukan ekstraksi nilai RGB yang kemudian akan di konversi ke nilai HSV. Dalam melakukan klasifikasi digunakan algoritma KNN, yang nilai terbaiknya ditentukan dengan nilai k , peneliti berhasil melakukan klasifikasi dengan nilai k terbaik dengan nilai akurasi yaitu 92% menggunakan 50 data citra uji, peneliti tidak menuliskan mengenai nilai k terbaik yang digunakan, sistem yang digunakan pun belum real time [12].

2.19.2 Implementasi *K-Nearest Neighbor* Untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB

Penelitian yang dilakukan oleh Lia Farokhah mengenai Implementasi K-Nearest Neighbor Untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB. Dalam penelitian ini dijelaskan bahwa KNN adalah metode klasifikasi paling sederhana yang dapat dengan mudah dipahami dan diimplementasikan, Peneliti ingin membuktikan mengenai kelebihan dan kekurangan KNN dalam proses ekstraksi warna pada objek gambar, objek yang digunakan pada penelitian ini adalah bunga *coltsfoot*, *daisy*, *dandelion* dan matahari. Hasil menunjukkan bahwa kolaborasi metode klasifikasi KNN dengan ekstraksi fitur warna RGB memiliki kelemahan terhadap percobaan pertama dengan akurasi 50-60% pada K=5. Percobaan kedua memiliki akurasi sekitar 90-100% pada K=5. Peningkatan akurasi, *precision* dan *recall* terjadi ketika menaikkan jumlah K yaitu dari K=1 menjadi K=3 dan K=5, peneliti menyimpulkan bahwa metode klasifikasi KNN kurang cocok untuk mengklasifikasikan objek dengan perbedaan warna yang mencolok [13].

2.19.3 Deteksi Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna HIS

Penelitian yang dilakukan oleh Rendy Pratama, Achmad Fuad Assagaf dan Firman Tempola membahas mengenai Deteksi Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna HIS. Dalam penelitian ini peneliti menggunakan objek buah tomat dengan nilai yang diukur dan diamati adalah nilai HIS (*Hue* , *Saturation* dan *Intensity*) dari buah tomat, objek buah tomat memiliki warna dan tentunya nilai HIS yang berbeda di setiap fase kematangan, Dimensi warna yang digunakan adalah HIS yang berbeda dengan RGB karena dapat melihat nilai intensitas cahaya pada objek, untuk dapat mendapatkan nilai HIS digunakan teknik transformasi warna dari pixel RGB, menggunakan HIS sebagai dimensi warna diperoleh hasil akurasi yaitu 94% [14].

2.19.4 Implementasi Deteksi Warna Pada Game *Finding Color* Menggunakan Ekstraksi Fitur Warna dan *Fuzzy Decision Tree*

Penelitian oleh Risna Hasanah, Eka Wahyu Hidayat dan Neng Ika Kurniati membahas mengenai Implementasi Deteksi Warna Pada *Game Finding Color* Menggunakan Ekstraksi Fitur Warna dan *Fuzzy Decision Tree*. Sistem yang digunakan di deploy pada *android* dan algoritma yang digunakan adalah *fuzzy decision tree*. Pada penelitian ini sistem adalah sebuah permainan anak yang dapat mengenali fitur warna pada piksel gambar yang disentuh, gambar yang tampil telah adalah gambar yang diambil secara langsung oleh pengguna, *fuzzy decision tree* digunakan untuk pengklasifikasian warna yang didapatkan hasil ujinya 100% akurat dengan 90 data tes, peneliti menyimpulkan *fuzzy decision tree* dapat digunakan dengan baik dalam pengklasifikasian warna jika tingkat pencahayaan baik dan normal [15].

2.19.5 Implementation of K-Nearest Neighbors Algorithm for Predicting Heart Disease Using Python Flask

Penelitian yang dilakukan oleh Dimas Aryo Anggoro dan Nur Chudlori Aziz membahas mengenai “*Implementation of K-Nearest Neighbors Algorithm for Predicting Heart Disease Using Python Flask*”. Algoritma yang digunakan adalah KNN, bahasa pengembangan adalah *Python* dan *framework* pengembangan adalah *Flask*, peneliti mengatakan bahwa *Flask* adalah *framework* pengembangan *website* khusus untuk bahasa *Python*, *Flask* juga tidak memiliki banyak *tool* dan *library* sehingga lebih *portable* dan tidak membutuhkan sumber daya yang banyak, peneliti menyimpulkan dalam pengembangan *machine learning* dengan *Python Flask* dapat menambah efisiensi kerja, peneliti juga menyimpulkan bahwa KNN dapat diandalkan dalam implementasi sistem berbasis web dengan *Flask* [16].

2.19.6 Sistem Pendeteksi Gejala Awal Covid-19 Dengan Penggunaan Metode AI Project Cycle

Penelitian Fauzan Azimah dan Kiky Rizky Nova Wardani mengenai Sistem Pendeteksi Gejala Awal Covid-19 Dengan Penggunaan Metode *AI Project Cycle*. Sistem yang dikembangkan pada penelitian ini dapat menghitung tingkat resiko

terjangkit covid-19 berdasarkan gejala yang timbul, Algoritma yang digunakan pada penelitian adalah *logistic regression*, *Random Forest Classifier* dan *SVM*, SDLC yang digunakan peneliti adalah *AI Project Cycle* yang memiliki 6 tahap, sistem juga di *deploy* secara *online* menggunakan *web hoster Heroku*, peneliti mengatakan *AI Project Cycle* adalah SDLC yang dalam tepat dalam pembuatan project berbasis *Artificial Intelligence*, peneliti juga berhasil mendapatkan tingkat akurasi 80% pada setiap algoritma yang digunakan [17].

2.19.7 Novice Programmer to New-Age Application Developer: What Makes Python their First Choice?

Penelitian yang dilakukan oleh Ramachandran Trichur Narayanan membahas mengenai “*Novice Programmer to New-Age Application Developer: What Makes Python their First Choice ?*” . Penelitian ini membahas mengenai alasan mengapa dan apa yang membelakangi para programmer pemula memilih *Python* sebagai bahasa pemrograman pertama mereka. Dijelaskan pada jurnal bahwa 3 bidang aplikasi *Python* yang paling umum adalah *web development*, *internet of things* dan *artificial intelligence*. Peneliti berargumentasi bahwa *Python* telah memiliki andil yang sangat signifikan dalam pengembangan *artificial intelligence* dengan kumpulan *librarynya*, fleksibilitas , skalabilitas serta *syntax* yang intuitif. Dengan jumlah *library* yang beragam yang mendukung perhitungan dan komputasi kompleks seperti *numpy*, *scipy*, *scikit*, *tensorflow* , *python* menjadi pilihan utama dalam pengembangan *artificial intelligence* dibandingkan bahasa lainnya seperti C dan Java, bahasa pemrograman lain yang dapat disandingkan dengan *Python* namun masih baru adalah R [20].

2.19.8 Klasifikasi Belimbing Menggunakan Naïve Bayes Berdasarkan Fitur Warna RGB

Penelitian yang dilakukan oleh Fuzy Yustika Manik dan Kana Saputra Saragih mengenai Klasifikasi Belimbing Menggunakan *Naïve Bayes* Berdasarkan Fitur Warna RGB membahas mengenai sistem penyortiran belimbing dengan ekstraksi fitur warna RGB pada gambar dengan algoritma *naïve bayes*. Dalam

penelitian didapatkan kesimpulan bahwa hasil klasifikasi menunjukkan akurasi menggunakan ekstraksi ciri RGB sebesar 80%. Penggunaan RGB sebagai ciri warna belum bisa digunakan sepenuhnya sebagai fitur dari citra buah belimbing. Hasil klasifikasi menggunakan algoritma *Naïve Bayes* masih terlalu rendah dibandingkan algoritma lain. Oleh karena itu, dibutuhkan ekstraksi ciri tambahan untuk mendukung model klasifikasi yang akan dibentuk.. Masih perlu ditambahkan ekstraksi fitur lainnya seperti tekstur, morfologi ataupun geometri [21].

2.19.9 Soil Color Detection Using Knn Classifier

Penelitian yang dilakukan oleh Shima Ramesh Maniyath, Akshatha K N, Architha L S, Mr. Ramachandra Hebbar dan Dr. S Rama Subramoniam membahas mengenai deteksi warna tanah menggunakan *KNN Classifier*. Penelitian yang dilakukan adalah mengenai warna tanah berdasarkan *munsell soil chart*, algoritma yang digunakan adalah KNN dan fitur warna yang digunakan adalah RGB dan HSV, *software* komputasi yang digunakan adalah Matlab dengan bahasa C/C++ , tidak disampaikan berapa angka keberhasilan klasifikasi namun peneliti menyatakan bahwa kode Matlab berhasil melakukan segmentasi nilai HSV dan *KNN Classifier* berhasil melakukan klasifikasi berdasarkan nilai RGB warna merujuk pada *munsell soil notation* [23].

2.19.10 Usability Testing Sesuai dengan ISO 9241-11 pada Sistem Informasi Program Pengalaman Lapangan Universitas Pendidikan Ganesha Ditinjau dari Pengguna Mahasiswa

Penelitian yang dilakukan oleh Ni Luh Surya Wardani, I Gede Mahendra Darmawiguna dan Nyoman Sugihartini membahas mengenai *Usability Testing* berdasarkan ISO 9421-11 pada Sistem Informasi Program Pengalaman Lapangan Universitas Pendidikan Ganesha Ditinjau dari Pengguna Mahasiswa . Pada penelitian ini dilakukan *Usability Testing* yang didasarkan dari standar ISO-9421-11 , di dalamnya dijelaskan mengenai tata cara pengujian sebuah aplikasi ataupun sistem dengan 3 faktor yang diukur yaitu efektifitas, efisiensi dan kepuasan, untuk

efektifitas dan efisiensi digunakan teknik *Performance Measusrement* dan untuk digunakan kuesioner SUS. Objek yang di teliti adalah Sistem Informasi Program Pengalaman Lapangan Universitas Pendidikan Ganesha hasil *Usability Testing* yang didapatkan adalah keefektifitasan menghasilkan 96,3% yaitu termasuk dalam kategori efektif., utuk efisiensi didapatkan masih belum efisien, dan untuk kepuasan mendapatkan skor 68 yang artinya responden belum puas [29].

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian dilakukan pada:

1. Waktu penelitian : Maret 2022 sampai dengan Juli 2022
2. Tempat penelitian : *Orbit Future Academy*

3.1.1 Jadwal Penelitian

Jadwal pada penelitian ini dapat dilihat pada Tabel 3.1 sebagai berikut:

Tabel 3. 1 Jadwal Penelitian

No	Aktivitas	Agust 2022	Sept 2022	Okt 2022	Nov 2022	Des 2022
1	<i>Problem Scoping</i>					
2	<i>Data Acquistion</i>					
3	<i>Data Exploration</i>					
4	<i>Modelling</i>					
5	<i>Evaluation</i>					
6	<i>Deploy & Finishing</i>					

3.2 Alat Dalam Penelitian

3.2.1 Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.2 sebagai berikut:

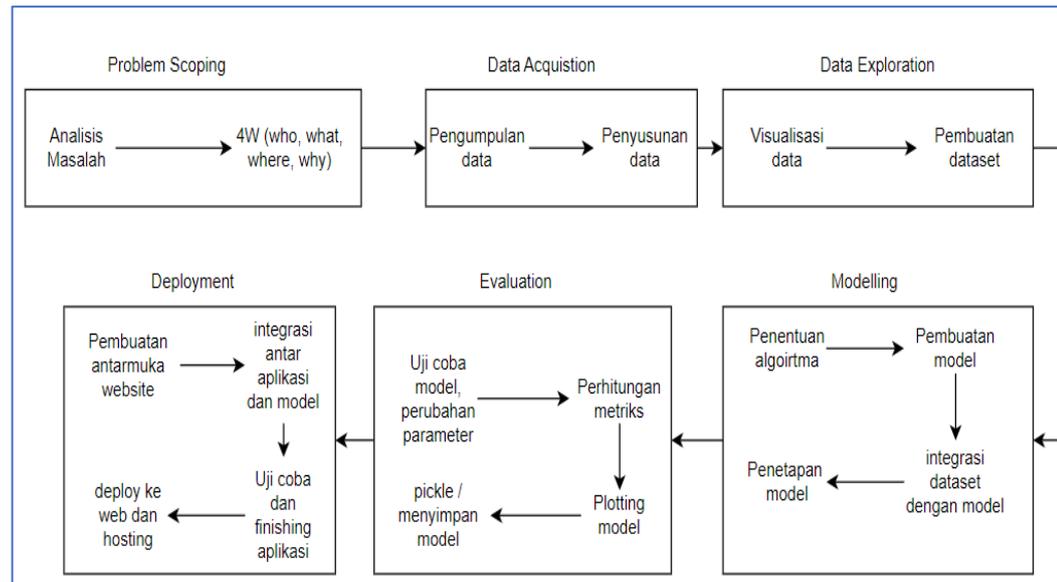
Tabel 3. 2 Alat yang Digunakan Dalam Penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	<i>Intel i5 11355G7, RAM 16 GB, dengan sistem operasi Windows 11</i>	Perangkat keras yang digunakan sebagai <i>compiler</i> dalam pemrograman
3	<i>Visual Studio Code</i>	-	<i>Integrated Development Environment (IDE)</i> untuk melakukan pengembangan aplikasi web
4	<i>Anaconda / Anaconda Prompt</i>	-	<i>Package ,Dependencies dan Environment manager</i> untuk <i>Python</i>
5	<i>Google Collab</i>	-	<i>IDE</i> untuk <i>training</i> model AI
6	<i>Figma</i>	-	Perangkat lunak untuk membuat prototype antarmuka web.
7	<i>Trello</i>	-	Perangkat lunak untuk dokumentasi.
8	<i>Git</i>	-	Perangkat lunak untuk dokumentasi
9	<i>Heroku</i>	-	Server <i>hosting</i>

3.2.2 Bahan Penelitian

Bahan yang digunakan dalam penelitian ini adalah contoh berbagai gambar.

3.3 Tahapan Penelitian



Gambar 3. 1 Tahapan Penelitian.

Gambar 3.1 adalah tahapan penelitian yang didasarkan pada AI Project Cycle kemudian dijabarkan lebih luas lagi sesuai kebutuhan. Proses pelaksanaan dengan judul penelitian “Aplikasi Web Pendeteksi Warna Pada *Pixel* Gambar Dengan KNN Classifier” adalah sebagai berikut. Proses ini diawali dengan *brainstorming* ide yang diputuskan untuk membawa ide *color detection* dengan domain *Computer Vision* (CV). Setelah ide ditentukan melanjutkan proses pengerjaan penelitian sesuai dengan *AI Project Cycle* [1].

3.3.1 Tahap Problem Scoping

3.3.1.1 Analisa Masalah

Setelah menentukan ide yang akan digunakan dalam penelitian ini maka langkah awal yang dilakukan adalah menganalisis masalah dan solusi yang

ditawarkan, pada *problem scoping* ini terdapat beberapa analisis masalah terkait ide proyek yang akan dibuat yakni *Color Detection* dengan rumusan 4W.

Kesimpulan dari rumusan ini diemukan beberapa analisa masalah terkait adalah sebagai berikut. Permasalahan utama pada proyek ini adalah untuk membuat suatu aplikasi yang dapat mendeteksi serta mengklasifikasi warna pada pixel gambar dikarenakan masalah utamanya adalah manusia memiliki keterbatasan dalam mendeteksi serta mengklasifikasi ratusan warna dalam gambar secara presisi. Selama ini juga banyak orang awam yang hanya mengetahui dan mendeteksi warna umum saja pada gambar. Masalah ini penting karena dengan adanya aplikasi inilah diberikan solusi untuk membantu pengguna agar dapat mendeteksi sekaligus mengklasifikasi warna pada suatu gambar yang ingin diketahui warnanya.

3.3.1.2 Analisa 4W

Berikut adalah penjabaran 4W dalam *problem scoping* penelitian ini :

- *Who*

1. Siapa saja yang mengalami masalah tersebut?

Semua orang yang mengalami kesulitan dalam mengklasifikasikan ratusan warna dan orang yang memiliki keterbatasan dalam membedakan warna, misalnya penderita buta warna.

- *What*

1. Apa masalah utamanya?

Rendahnya pengetahuan terkait macam-macam warna dalam jumlah banyak, serta kesulitan dalam membedakan dan mengklasifikasikan ratusan warna.

2. Apa faktor-faktor pendukung masalah tersebut?

Penggunaan alat pendeteksi warna yang mudah dan keterbatasan sarana untuk mengetahui warna.

- *Where*

1. Dimana saja si calon pengguna mengalami masalah ini?

Dimana saja ketika orang ingin mengklasifikasikan warna, contohnya seperti :

di bidang kesenian, misalnya seniman yang kesulitan dalam menentukan warna dan di bidang tekstil, misalnya kesulitan menentukan warna kain.

- *Why*

1. Mengapa masalah ini penting?

Untuk mengatasi permasalahan seseorang yang kesulitan dalam mengklasifikasikan dan membedakan warna.

2. Apa nilai utama dari permasalahan ini?

Nilai utama dari permasalahan ini yaitu memberikan wawasan yang luas terkait macam-macam warna serta memudahkan pengguna dalam mengklasifikasikan dan membedakan warna.

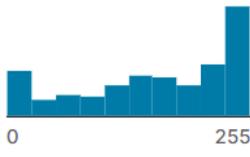
3. Bagaimana kamu bisa memperbaiki (sebagian atau seluruh) dari masalah tersebut?

Dengan mengembangkan teknologi dengan membuat aplikasi pendeteksi warna yang mudah untuk digunakan.

3.3.2 Tahap *Data Acquisition*

Setelah menganalisis masalah dan menemukan solusi untuk menyelesaikan masalah tersebut, langkah selanjutnya ialah proses *data acquisition* yang mana ini merupakan proses pengukuran, pengumpulan dan validasi data yang dibutuhkan dalam penelitian ini. proses ini diawali dengan membuat list data apa saja yang dibutuhkan. Data yang dibutuhkan antara lain adalah : nama warna, nilai RGB (*Red, Green, Blue*) dan Nilai *Hexadecimal* warna tersebut. setelah itu dilakukan proses *data acquisition* dengan mengambil data yang sudah tersedia di internet atau web *open source* seperti *kaggle*, *data.go.id*, *github* dan *gitlab*. kemudian mendownload dataset serta mengumpulkannya menjadi suatu format *.csv file*.

3.3.3 Tahap *Data Exploration*

▲ Name	▲ Hex (24 bit)	# Red (8 bit)
color name	hexadecimal way to represent a color in RGB format by combining three values	8 bits for each red
1291 unique values	1156 unique values	
Absolute zero	#0048BA	0
Acid green	#B0BF1A	176
Aero	#7CB9E8	124
Aero blue	#C9FFE5	201
African violet	#B284BE	178
Air Force blue (RAF)	#5D8AA8	93
Air Force blue (USAF)	#00308F	0

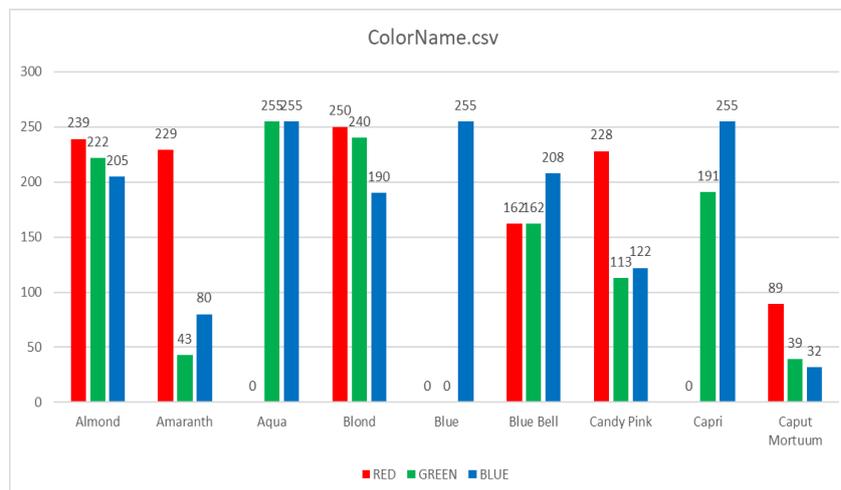
Gambar 3. 2 Contoh Data Pada *Website Kaggle*.

Gambar 3.2 diatas adalah contoh bentuk data mentah pada *website kaggle*, data dapat diambil dari berbagai sumber atau *website* yang menyediakan *dataset* atau data mentah gratis yang paling umum adalah dari *kaggle* dan *github*. Eksplorasi data dilakukan secara manual menggunakan *tools* dan aplikasi pengolah data mentah seperti *Microsoft Excel* dan *tableau*.

A	B	C	D	E
timetweet	tweet	id	name	label
4/6/2022 17:11	b'Pengumuman,\nDatang hari	1.51E+18	LDyantomy	0
4/6/2022 17:10	b'RT @LDyantomy: (DonetskP	1.51E+18	syafrijamal	0
4/6/2022 17:09	b'RT @Paltiwest: Wajarlah Pre	1.51E+18	SRufamin	0
4/6/2022 17:08	b'RT @yo2thok: "Peta masa d	1.51E+18	inyonkjm	0
4/6/2022 17:08	b'RT @YudhoyonoInst: Bahas	1.51E+18	m_natsirdm	1
4/6/2022 17:05	b'RT @VIVAcoid: Astaga, PBB	1.51E+18	TjahjoWidia	0
4/6/2022 17:05	b'Volodymr Zelensky mengata	1.51E+18	detikcom	1
4/6/2022 17:05	b'(Jerman dalam pembicaraan	1.51E+18	LDyantomy	1
4/6/2022 17:05	b'@lachusnaf Bantuin nyelesa	1.51E+18	iniakunsoda	1
4/6/2022 17:04	b'Inggris: Dunia harus bertinda	1.51E+18	ElshintaTeg	1
4/6/2022 17:04	b'makin panjang garis komunil	1.51E+18	ketertibanu	0
4/6/2022 17:03	b'RT @Paltiwest: Wajarlah Pre	1.51E+18	rakyat_pem	0
4/6/2022 17:01	b'RT @privetbro1: Kementerie	1.51E+18	SetiadiYudiz	0
4/6/2022 17:00	b'@yo2thok Pro ukraina,, bias	1.51E+18	saiful_pulpu	0
4/6/2022 17:00	b'RT @Naga_Wicaksana2: Sep	1.51E+18	Naga_Wical	0

Gambar 3. 3 Contoh *Dataset*

Gambar 3.3 adalah tampilan sebuah *dataset* yang telah diolah di *microsoft excel*. Data yang telah dikumpulkan pada tahap sebelumnya nantinya akan disusun dan diolah sehingga dapat dibaca dan dipahami, gambar diatas adalah contoh *dataset tweet* pada aplikasi *twitter*, data bisa diolah jika diperlukan seperti *cleaning* maupun *labelling*, dataset nantinya akan disimpan dalam format csv atau txt.



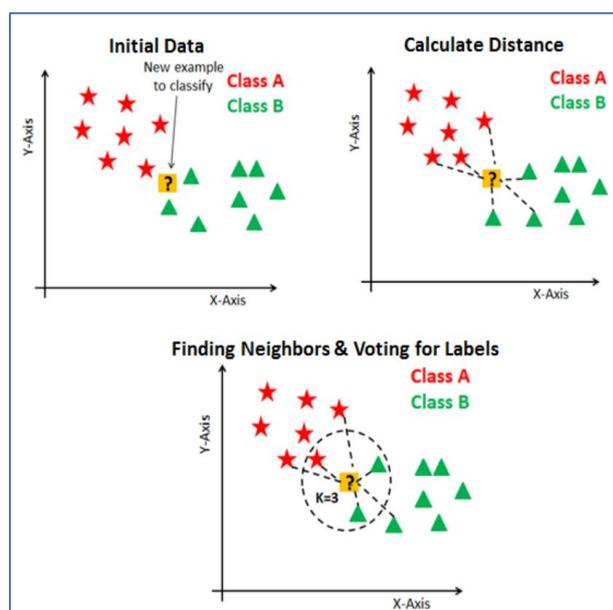
Gambar 3. 4 Contoh Visualisasi Data

Gambar 3.4 diatas adalah contoh visualisasi data menggunakan aplikasi *tableau*, pada aplikasi dapat dilihat *trend* data, jenis data dan kombinasi kombinasi

data yang sudah buat dalam *dataset*, sehingga data dapat lebih dipahami untuk pembuatan model.

3.3.4 Tahap *Modelling*

Selanjutnya ialah proses *modelling*, pada tahap ini adalah menentukan algoritma serta metode yang akan digunakan pada *project*. Dikarenakan *project* merupakan pengklasifikasian warna, oleh karena itu model yang digunakan ialah dalam kategori algoritma *classification* dengan metode *K-Nearest Neighbor*. Algoritma ini digunakan dalam menentukan nilai terdekat dari nilai *input* RGB pada gambar yang di *upload*, nilai terdekat yang telah ditentukan oleh *user* akan dideteksi serta diklasifikasikan sesuai dengan nilai pada *dataset*. sehingga menghasilkan suatu *output* berupa nama warna yang sesuai dengan nilai terdekat tersebut.



Gambar 3. 5 Klasifikasi dengan metode K-NN [23]

Gambar 3.5 adalah alur klasifikasi menggunakan KNN yang sebelumnya sudah dibahas pada bab II. Metode K-NN dimulai dengan terlebih dahulu menghitung jarak satu pengamatan uji dari semua pengamatan yang ada pada *dataset* latih dan kemudian dilanjutkan dengan mencari K (n) tetangga terdekatnya. Langkah di atas dilakukan untuk setiap pengamatan uji dan ini adalah bagaimana

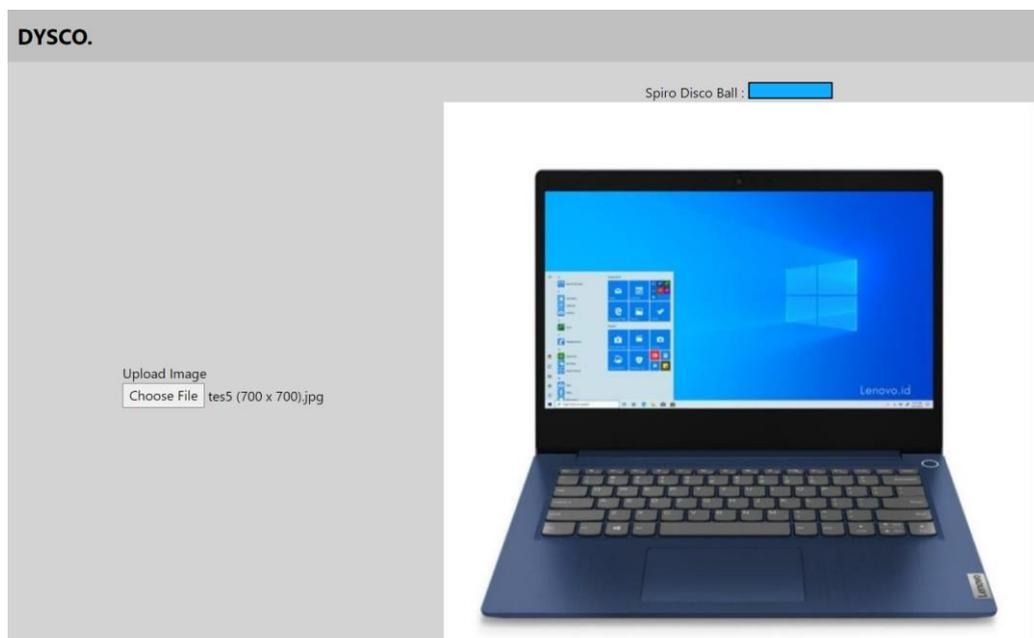
kesamaan yang ada dalam data ditemukan. Untuk membuat algoritma bekerja dengan baik pada kumpulan data, diperlukan metrik jarak yang paling sesuai.

3.3.5 Tahap Evaluation

Pada proses evaluasi merupakan proses dimana dilakukan pencarian model terbaik yang sesuai dengan proyek yang telah dibuat modelnya, serta menghitung nilai keakuratan dari model tersebut. Perhitungan evaluasi akan dilakukan menggunakan *metrics accuracy, precision dan recall*.

3.3.6 Tahap Deployment

Terakhir adalah proses *deployment*. *Deployment* aplikasi dilakukan dengan tujuan memudahkan penggunaan aplikasi dan kustomisasi *User Interface* (UI) aplikasi. *Deployment* aplikasi dilakukan menggunakan kerangka kerja *Flask*. *Flask* adalah kerangka kerja digunakan untuk penanganan *backend* (*server-side*) aplikasi. pada aplikasi ini *Flask* digunakan untuk menghubungkan antara *View* (file HTML, CSS, dan JS) dan *Model* (file Python dan Dataset).



Gambar 3. 6 Contoh Hasil Akhir Tampilan

Gambar 3.6 adalah contoh tampilan UI minimum yang diinginkan. *Deployment* aplikasi akan dilakukan secara lokal dengan web *browser* maupun *online* menggunakan platform HEROKU.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan dapat diambil kesimpulan bahwa :

1. Pembuatan aplikasi web pendeteksi warna pada *pixel* gambar dengan KNN *Classifier* telah berhasil dilakukan. Aplikasi web sudah menggunakan KNN *Classifier* sebagai algoritma utamanya. Aplikasi web juga sudah mampu mendeteksi dan memberikan informasi warna pada gambar, dengan akurasi yang tinggi yaitu 0,84.
2. Aplikasi web sudah dapat memenuhi fungsi fungsi utamanya, ditunjukkan melalui pengujian menggunakan teknik *Usability Testing* berdsarkan ISO 9421-11 untuk mengukur *Efectivity*, *Eficiency* dan *Satisfaction*, dengan 10 orang responden awam. Hasil *Usability Testing* mendapatkan skor yang sangat bagus dengan nilai *efectivity* yaitu 100% dengan predikat sangat efektif, nilai *eficiency* yaitu 32,6 detik dengan predikat sangat cepat, dan nilai *satisfaction* yaitu 84 dengan predikat dapat diterima (*acceptable*).

5.2 Saran

Saran yang didapatkan dari penelitian ini adalah sebagai berikut :

1. Perlu adanya penilitian lebih lanjut agar aplikasi dapat dioperasikan dalam bentuk *realtime*.
2. Dalam pengembangan aplikasi kedepannya, diharapkan aplikasi dapat berbasis *Android*.
3. Diharapkan pengembangan dataset warna yang digunakan dalam penamaan warna dapat lebih sederhana dan berbahasa indonesia.
4. Dalam pembuatan model selanjtnya, diharapkan dapat menggunakan algoritma lain yang lebih akurat.

DAFTAR PUSTAKA

- [1] CBSE Academic. “*AI Curriculum Handbook*” . 2019. https://cbseacademic.nic.in/web_material/Curriculum20/AI_Curriculum_Handbook.pdf (accessed on Jul 21, 2021)
- [2] Wibowo, W., Ulama, B. S. S., & Azies, H. Al. . “Belajar Pemrograman Bahasa Python(Edisi Pert)”. ITS Press. 2020.
- [3] Grinberg, M. . “Flask Web Development(Edisi Pert)”. O’Reilly Media, Inc. 2014
- [4] Robin Nixon. “Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition”. O’Reilly Media, Inc. 2014.
- [5] Atlassian, “About , What is Trello?,” 2021. <https://trello.com/> (accessed on Jul. 21, 2021).
- [6] Alessandro Del Sole. “ Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux”. Apress Media, LLC. 2019.
- [7] John Paul Mueller. “ Beginning Programming with Python® For Dummies®, 2nd Edition” . John Wiley & Sons, Inc . 2018.
- [8] Google, “Google Colab”, 2022. <https://research.google.com/colaboratory/faq.html> (accessed Nov. 7, 2021).
- [9] Anubhav Hanjura. “ Heroku Cloud Application Development”. Packt Publishing Ltd.. 2014.
- [10] Risa H. ,Risky A. R. ,Siti R.. “DATA MINING, K-Nearest Neighbor”. Fakultas Teknik Universitas Nusantara PGRI. 2017
- [11] Saha, Avishek , “Color Names data”, 2022, <https://www.kaggle.com/datasets/avi1023/color-names> (accessed Sept. 12 , 2022)

- [12] Syahid, Dani. dkk. “Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (Knn) Berdasarkan Nilai Hue, Saturation, Value (Hsv)”. JOIN. 2016
- [13] Farokhah, Lia . “ Implementasi K-Nearest Neighbor Untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna Rgb”. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK). Vol. 7 . 2020
- [14] Pratama, Rendy. dkk. “Deteksi Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna His”. JIKO (Jurnal Informatika dan Komputer). Vol. 2. 2019
- [15] Hasanah, Risna. dkk. “Implementasi Deteksi Warna Pada Game Finding Color Menggunakan Ekstraksi Fitur Warna dan Fuzzy Decision Tree” . Jurnal Teknik Informatika dan Sistem Informasi. Vol. 6. 2020
- [16] Anggoro, D. A. , Aziz, N. C.. “ Implementation of K-Nearest Neighbors Algorithm for Predicting Heart Disease Using Python Flask” . Iraqi Journal of Science. 2020
- [17] Azimah, Fauzan. dkk . “Sistem Pendeteksi Gejala Awal Covid-19 Dengan Penggunaan Metode Ai Project Cycle” . The Fourth Bina Darma Conference on Computer Science (BDCCS2022). 2022
- [18] Susilo, Edi. “Cara Menggunakan System Usability Scale (SUS) Pada Evaluasi Usability”. 2019 . <https://www.edisusilo.com/cara-menggunakan-system-usability-scale> (accessed on dec. 2022)
- [19] Z. Sharfina and H. B. Santoso, “An Indonesian adaptation of the System Usability Scale (SUS),” in International Conference on Advanced Computer Science and Information Systems, ICACISIS 2016, 2017, pp. 145–148.
- [20] R. T. Narayanan. “Novice Programmer to New-Age Application Developer: What Makes Python their First Choice?” . 10th ICCCNT. 2019
- [21] Y. M. , Fuzy , dkk. “Klasifikasi Belimbing Menggunakan Naïve Bayes Berdasarkan Fitur Warna RGB”. IJCCS Vol.11. 2017
- [22] Aiforkids . “Domains of AI Class 10 CBSE “.2022. <https://aiforkids.in/class-10/domains-of-ai/> (accessed on Jan. 15, 2023)
- [23] Ertel, Wolfgang, “Introduction to Artificial Intelligence Second Edition”. Springer International Publishing. 2017
- [24] R. M., Shima, dkk. “Soil Color Detection Using Knn Classifier”. 8 International Conference on Design Innovations for 3Cs Compute Communicate Control. 2018

- [25] ISO. “International Organization for Standardization (ISO) 9241-11:2018”. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> (Accessed on Jan. 15, 2023)
- [26] Brooke, J. “SUS: a retrospective. Journal of Usability Studies”. 2013
- [27] Tuloli , Mohamad Syafri , dkk. “Pengukuran Tingkat Usability Sistem Aplikasi e-Rapor Menggunakan Metode Usability Testing dan SUS”. Jambura Journal Of Informatics Vol. 4. 2022
- [28] Ayu , Putu Desiana Wulaning . “Analisis Pengukuran Tingkat Efektivitas dan Efisiensi Sistem Informasi Manajemen Surat STIKOM Bali”. Jurnal Sistem Dan Informatika Vol. 11. 2017
- [29] Wardani, Ni Luh Surya . “Usability Testing Sesuai dengan ISO 9241-11 pada Sistem Informasi Program Pengalaman Lapangan Universitas Pendidikan Ganesha Ditinjau dari Pengguna Mahasiswa”. Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI) Volume 8. 2019
- [30] Pengo. “Pixel Geometries”. 2012. https://en.wikipedia.org/wiki/Pixel#/media/File:Pixel_geometry_01_Pengo.jpg . (Accessed on Feb. 15 ,2023)
- [31] Lyon, Richard F. “A brief history of pixel” . 2006 . <http://www.dicklyon.com/tech/Photography/Pixel-SPIE06-Lyon.pdf>. (Accessed on Feb. 15, 2023)
- [32] Singh ,Himanshu . “Practical Machine Learning and Image Processing”. Springer Science Business Media New York. 2019
- [33] L. Tanimoto, Steven . “AN INTERDISCIPLINARY INTRODUCTION TO IMAGE PROCESSING Pixels, Numbers, and Programs”. Massachusetts Institute of Technology. 2012
- [34] Madenda, Sarifuddin. “ Pengolahan Citra dan Video Digital”. Penerbit Erlangga. 2015
- [35] Athuraliya ,Amanda. “ The Easy Guide to Component Diagrams “. 2023 . <https://creately.com/blog/software-teams/component-diagram-tutorial/> . (Accessed on Feb. 15, 2023)