

BAB I

PENDAHULUAN

1.1 Latar belakang

Kemajuan teknologi informasi dan komunikasi yang semakin pesat telah mendorong meningkatnya penggunaan data digital, khususnya dalam bentuk citra atau gambar digital. Citra digital saat ini dimanfaatkan di berbagai bidang, mulai dari media sosial, fotografi, kesehatan, keamanan, hingga sistem pengenalan wajah dan pengawasan berbasis visual. Perkembangan perangkat digital seperti kamera beresolusi tinggi dan smartphone modern semakin mendorong kebutuhan akan citra berkualitas tinggi. Namun, citra beresolusi tinggi menghasilkan ukuran file yang besar, yang menimbulkan permasalahan seperti kebutuhan ruang penyimpanan yang lebih besar, lambatnya proses transmisi data, serta menurunnya efisiensi pemrosesan citra secara real-time.

Salah satu solusi untuk mengatasi permasalahan tersebut adalah dengan menerapkan kompresi citra digital. Kompresi citra merupakan proses pengurangan jumlah data yang dibutuhkan untuk merepresentasikan citra tanpa menghilangkan informasi penting. Tujuannya adalah memperkecil ukuran file, mempercepat pengiriman data, serta meningkatkan efisiensi pemrosesan. Terdapat dua pendekatan utama dalam kompresi citra, yaitu lossless compression (tanpa kehilangan data) dan lossy compression (dengan kehilangan sebagian data). Dalam praktik multimedia, metode lossy lebih banyak digunakan karena mampu mencapai rasio kompresi yang tinggi dengan kualitas visual yang masih dapat diterima.

Salah satu metode lossy yang populer adalah Discrete Cosine Transform (DCT). Metode ini mengubah citra dari domain spasial ke domain frekuensi, memisahkan komponen frekuensi rendah dan tinggi. Komponen penting citra terdapat pada frekuensi rendah yang dipertahankan, sedangkan detail halus pada frekuensi tinggi dapat dikurangi. Teknik ini banyak digunakan pada standar kompresi JPEG karena dapat menghasilkan ukuran file lebih kecil dengan kualitas yang relatif baik.

Namun demikian, penerapan metode DCT pada aplikasi kompresi citra masih menghadapi beberapa tantangan. Pertama, diperlukan penentuan parameter nilai kompresi yang tepat agar hasil optimal. Kedua, hasil kompresi sering kali berbeda pada setiap citra meskipun menggunakan parameter yang sama, sehingga perlu kajian lebih mendalam. Ketiga, perlu diuji apakah citra hasil kompresi masih dapat dikompresi kembali (re-compression) secara signifikan tanpa menurunkan kualitas berlebihan. Keempat, aplikasi sebaiknya mendukung berbagai format citra, tidak hanya JPEG dan PNG, tetapi juga format lain seperti GIF dan BMP. Kelima, hasil pengujian perlu dilengkapi dengan kolom persentase selisih ukuran file untuk mengukur efektivitas kompresi secara lebih jelas.

Penelitian terdahulu menunjukkan bahwa metode DCT efektif dalam mengurangi ukuran file citra. Misalnya, penelitian oleh Suwardoyo & Dwiyantri (2023) menunjukkan bahwa penerapan DCT pada kompresi citra JPEG mampu mengurangi ukuran file hingga 60% dengan kualitas visual yang tetap layak digunakan. Penelitian lain oleh Yunita (2022) membandingkan DCT dengan metode Discrete Wavelet Transform (DWT) dan menyimpulkan bahwa DCT lebih unggul dalam kecepatan kompresi, meskipun kualitas citra DWT sedikit lebih baik

pada tingkat kompresi tinggi. Sementara itu, **Sari (2021)** merancang aplikasi kompresi berbasis DCT namun hanya mendukung format JPEG dan PNG tanpa menyediakan opsi re-compression.

Berdasarkan celah tersebut, penelitian ini menghadirkan pembaruan dengan cara: (1) menambahkan dukungan format gambar lain seperti GIF dan BMP, (2) menampilkan parameter nilai kompresi secara eksplisit, (3) menguji kemampuan aplikasi dalam melakukan re-compression, serta (4) menyajikan hasil uji dengan tambahan kolom persentase selisih ukuran file. Dengan demikian, penelitian ini diharapkan mampu memberikan kontribusi baru dalam pengembangan aplikasi kompresi citra berbasis metode DCT yang lebih komprehensif dan aplikatif.

Berdasarkan latar belakang diatas maka penulis memiliki beberapa gagasan untuk menggangakat judul skripsi ini :”**Aplikasi Kompresi Citra Menggunakan Metode *Discrete Cosine Transform (DCT)***”.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diidentifikasi dalam penelitian ini adalah:

1. Bagaimana merancang aplikasi kompresi citra digital menggunakan metode Discrete Cosine Transform (DCT) dengan dukungan berbagai format gambar (JPEG, PNG, GIF, dan BMP)?
2. Bagaimana menentukan dan menampilkan parameter nilai kompresi pada aplikasi sehingga dapat dianalisis perbedaan ukuran file dan kualitas hasil kompresi?

3. Bagaimana kemampuan aplikasi dalam melakukan kompresi ulang (re-compression) terhadap citra hasil kompresi sebelumnya agar tetap signifikan dalam pengurangan ukuran file?

1.3 Batasan Masalah

Adapun batasan masalah adalah sebagai berikut :

1. Penelitian hanya membahas kompresi citra digital berformat JPEG, PNG GIF, dan BMP.
2. Metode yang digunakan dalam proses kompresi menggunakan *discrete cosine transform* (DCT) tanpa kombinasi metode lain.
3. Perangkat lunak aplikasi kompresi citra dibangun dengan menggunakan bahasa pemogramman HTML, PHP dan Javascript berbasis website.

1.4 Tujuan Penelitian

Tujuan Penelitian skripsi ini adalah sebagai berikut :

1. Merancang dan membangun aplikasi kompresi citra digital berbasis metode *discrete cosine transform* (DCT).
2. Menganalisis performa metode DCT dalam meningkatkan rasio kompresi citra digital.
3. Memberikan solusi efisien untuk penyimpanan dan transmisi citra digital melalui aplikasi kompresi yang efektif dan mudah digunakan.

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah:

1. Manfaat Teoritis:
 - a. Menambah wawasan dan referensi ilmiah terkait penerapan metode *discrete cosine transform* (DCT) dalam bidang kompresi citra digital.
 - b. Memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang pengolahan citra dan teknik kompresi.
2. Manfaat Praktis:
 - a. Memberikan solusi praktis dalam mengurangi ukuran file citra digital.
 - b. Aplikasi yang dibangun dapat digunakan oleh pengguna umum, peneliti, maupun pelajar untuk memahami konsep dan praktik kompresi citra secara langsung.
 - c. Membantu efisiensi penggunaan media penyimpanan dan bandwidth jaringan dalam pengiriman gambar digital.

1.6 Sistematika Penulisan

Adapun untuk sistematika penulisan tugas akhir ini terdiri dari 5 bab, yaitu:

BAB I : PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metode penelitian dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi tentang kajian teori-teori yang digunakan sebagai landasan dalam penelitian dan pengertian aplikasi yang digunakan.

BAB III : METODE PENELITIAN

Bab ini membahas langkah dari proses pembuatan analisa dan perancangan sistem dan perancangan antarmuka sistem kompresi citra dan metode *discrete cosine transform* (DCT).

BAB IV : HASIL PENELITIAN

Bab ini menunjukkan hasil disertai dengan analisa sehingga didapatkan bukti kuat dari penelitian skripsi yang dilakukan.

BAB V : KESIMPULAN DAN SARAN

Bab ini menyimpulkan apa yang ada pada bab-bab terdahulu serta memberikan saran atas penulisan skripsi ini.

BAB II

TINJAUAN PUSTAKA

2.1 Citra

Citra hasil rekaman kamera digital sering sekali terdapat beberapa gangguan yang mungkin terjadi, seperti lensa tidak fokus, muncul bintik-bintik yang disebabkan oleh proses capture yang tidak sempurna, pencahayaan yang tidak merata yang mengakibatkan intensitas tidak seragam, kontras citra terlalu rendah sehingga objek sulit dipisahkan dari latarbelakangnya, atau gangguan yang disebabkan oleh kotoran- kotoran yang menempel pada citra dan lain sebagainya.

Citra (*image*) atau istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Meskipun sebuah citra kaya akan informasi, namun sering kali citra yang dimiliki mengalami penurunan mutu, misalnya mengandung cacat atau *noise* (Fanggidae et al., 2020). Tentu saja citra semacam ini menjadi lebih sulit untuk diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Untuk mengatasi *noise* tersebut perlu dilakukan usaha untuk memperbaiki kualitas citra itu.

2.1.1 Foto

Generasi muda yang telah akrab dengan dunia digital atau lebih dikenal sebagai Digital native atau generasi milenial, belajar, bermain dan bersosialisasi dengan menggunakan media *online*. Pemberitaan dengan menggunakan unsur foto jurnalistik pada media online berperan untuk memengaruhi para pembaca, agar

keaktualan berita dapat dipercaya terhadap peneguhan pengetahuan dari pengalaman yang telah diperoleh sebelumnya, menambah wawasan pengetahuan, dan membentuk opini. Foto jurnalistik mampu merekam sesuatu secara cepat, objektif, hingga mengandung berita dan informasi. Dengan foto jurnalistik, isi dari berita dapat tersirat terlebih dahulu.

Besarnya pengaruh foto dalam media membuat peranan dari foto semakin tinggi dalam menyampaikan informasi kepada masyarakat secara luas dan akurat. Media foto jurnalistik merupakan salah satu komunikasi lewat foto. Foto jurnalistik adalah foto yang dimuat dalam suatu media, melengkapi suatu berita, artikel, dan memiliki nilai berita atau menjadi berita itu sendiri (Romadhoni, 2023).

2.2 Pengolahan Citra

Secara matematis, citra merupakan fungsi kontinu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel= picture element*) atau elemen terkecil dari sebuah citra. (Ria et al., 2022).

$$f(x,y)= \begin{bmatrix} f(0,0) & f(0,1) & f(0, M - 1) \\ f(1,0) & f(1,1) & f(1, M - 1) \\ \vdots & \vdots & \vdots \\ f(N - 1,0) & f(N - 1,1) & f(N - 1, M - 1) \end{bmatrix} \dots\dots\dots (2)$$

Suatu citra $f(x,y)$ dalam fungsi matematis dapat dituliskan sebagai berikut:

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$0 \leq f(x,y) \leq G-1$$

Keterangan

M = jumlah piksel baris (*row*) pada array citra

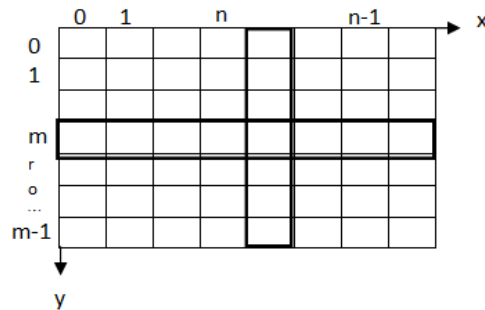
N = jumlah piksel kolom (*column*) pada array citra

G = nilai skala keabuan (*graylevel*)

Besarnya nilai M , N dan G pada umumnya merupakan perpangkatan dari dua.

$$M = 2^m ; N = 2^n ; G = 2^k \dots\dots\dots (3)$$

Interval $(0,G)$ disebut skala keabuan (*grayscale*). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1(satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan)



Gambar 2. 1 Representasi Citra Digital Dalam 2D

(Sumber : Ria et al., 2022).

2.3 Kompresi Citra

Kompresi citra adalah suatu proses yang bertujuan untuk mengurangi jumlah data yang dibutuhkan untuk merepresentasikan suatu citra digital. Dalam dunia digital yang berkembang pesat, penyimpanan dan transmisi data visual seperti gambar dan video memerlukan ruang dan *bandwidth* yang besar. Oleh karena itu,

teknik kompresi citra sangat penting dalam mengoptimalkan efisiensi penggunaan media penyimpanan dan mempercepat proses pengiriman data tanpa mengorbankan kualitas secara signifikan. Kompresi citra menjadi salah satu elemen penting dalam pengolahan citra digital, terutama pada sistem komunikasi, multimedia, aplikasi berbasis web, perangkat seluler, hingga pengarsipan citra medis (Pangesti et al., 2020).

Secara umum, kompresi citra dapat dibedakan menjadi dua kategori utama, yaitu kompresi *lossless* dan *lossy*. Kompresi *lossless* adalah metode kompresi yang tidak menyebabkan kehilangan informasi pada citra asli. Setelah proses dekompresi, citra yang dihasilkan akan identik dengan citra sebelum dikompresi. Metode ini sangat berguna dalam aplikasi yang membutuhkan akurasi tinggi, seperti dalam bidang kedokteran, arsip digital, atau pengolahan citra teknik. Beberapa algoritma yang sering digunakan dalam kompresi *lossless* antara lain adalah *Run-Length Encoding (RLE)*, *Huffman Coding*, dan *Lempel-Ziv-Welch (LZW)*. *Run-Length Encoding* bekerja dengan cara mengompresi data berulang secara efisien, sementara *Huffman Coding* menggunakan frekuensi kemunculan simbol untuk membentuk kode biner terpendek bagi simbol yang paling sering muncul. LZW, yang merupakan dasar dari format GIF, melakukan kompresi dengan menggabungkan pola berulang dalam citra menjadi satu entitas.

Berbeda dengan *lossless*, kompresi *lossy* memungkinkan pengurangan data citra dengan cara menghilangkan sebagian informasi yang dianggap tidak terlalu penting oleh sistem persepsi visual manusia. Akibatnya, hasil dekompresi dari citra tidak identik dengan citra asli, namun tetap terlihat cukup mirip secara visual. Metode ini umumnya digunakan pada aplikasi yang memerlukan efisiensi tinggi

seperti dalam penyimpanan foto digital (JPEG), video streaming (MPEG), dan platform media sosial. Salah satu teknik yang paling umum digunakan dalam kompresi *lossy* adalah *Discrete Cosine Transform* (DCT). DCT memproyeksikan data citra ke dalam domain frekuensi sehingga memungkinkan identifikasi bagian-bagian citra yang bisa dikompresi secara efisien. Selain DCT, metode lain seperti transformasi *wavelet* dan standarisasi JPEG2000 juga digunakan dalam sistem kompresi citra modern karena kemampuannya dalam mempertahankan kualitas visual dengan ukuran file yang kecil.

Dalam implementasinya, kualitas kompresi citra tidak hanya diukur dari seberapa kecil ukuran file yang dihasilkan, tetapi juga dari seberapa baik citra hasil dekompresi dalam mempertahankan kemiripan visual dengan citra aslinya. Beberapa parameter evaluasi umum digunakan untuk menilai kinerja sistem kompresi citra, antara lain *Compression Ratio* (CR), *Peak Signal-to-Noise Ratio* (PSNR), dan *Mean Squared Error* (MSE). *Compression Ratio* menunjukkan perbandingan ukuran citra asli dengan ukuran citra yang telah dikompresi. Semakin tinggi nilai CR, maka semakin efisien metode kompresi tersebut. PSNR digunakan untuk mengukur tingkat kejernihan citra hasil kompresi, dan nilai yang tinggi menunjukkan kualitas yang lebih baik. MSE mengukur rata-rata kesalahan kuadrat antara piksel citra asli dengan hasil dekompresi, dan semakin rendah nilai MSE maka semakin baik kualitas kompresi (Sugiyono, 2022).

2.4 Metode discrete cosine transform (DCT)

Discrete Cosine Transform (DCT) merupakan salah satu metode transformasi frekuensi yang paling banyak digunakan dalam proses kompresi citra, terutama

dalam sistem kompresi berbasis *lossy* seperti standar JPEG. DCT bekerja dengan cara mengubah representasi spasial suatu citra (berbasis piksel) menjadi representasi dalam domain frekuensi. Transformasi ini memungkinkan pemisahan informasi citra berdasarkan tingkat kepentingannya, sehingga informasi yang dianggap kurang penting (biasanya frekuensi tinggi) dapat dibuang untuk menghemat ruang penyimpanan tanpa memengaruhi kualitas visual secara signifikan.

DCT pertama kali diperkenalkan oleh Ahmed, Natarajan, dan Rao pada tahun 1974, dan sejak saat itu telah menjadi tulang punggung dalam berbagai teknik kompresi multimedia. Keunggulan utama dari DCT adalah kemampuannya untuk memusatkan energi sinyal pada koefisien frekuensi rendah. Hal ini membuat sebagian besar informasi penting dari citra terkonsentrasi hanya pada beberapa koefisien pertama, sedangkan koefisien lainnya bisa diabaikan atau dikompresi secara agresif (Suwardoyo & Dwiyantri, 2023).

Secara matematis, DCT satu dimensi untuk deret data dengan panjang N dapat dinyatakan dengan rumus:

$$X_k = \alpha(k) \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right], \quad k = 0, 1, \dots, N - 1$$

dengan:

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{jika } k = 0 \\ \sqrt{\frac{2}{N}} & \text{jika } k > 0 \end{cases}$$

Untuk pengolahan citra dua dimensi (2D), rumus DCT diperluas menjadi:

$$F(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right]$$

dimana:

1. $f(x,y)$ adalah nilai intensitas piksel pada koordinat (x,y) ,
2. $F(u,v)$ adalah koefisien hasil transformasi DCT pada koordinat frekuensi (u,v)
3. N adalah ukuran blok citra (biasanya 8×8 piksel),
4. $\alpha(u)$ dan $\alpha(v)$ adalah faktor normalisasi seperti pada DCT satu dimensi.

Dalam proses kompresi citra JPEG, DCT digunakan dalam blok-blok citra berukuran 8×8 piksel. Setiap blok mengalami transformasi DCT dua dimensi untuk menghasilkan koefisien frekuensi. Koefisien ini kemudian dikuantisasi (*quantization*) menggunakan tabel kuantisasi standar. Proses kuantisasi mengurangi presisi dari koefisien tertentu, terutama frekuensi tinggi yang dianggap kurang penting secara visual, sehingga menghasilkan penghematan data. Setelah dikuantisasi, koefisien tersebut kemudian dikodekan menggunakan metode kompresi entropi seperti *Huffman Coding* atau *Run-Length Encoding* (RLE) untuk menghasilkan bitstream akhir.

Kelebihan DCT adalah efisiensi energinya yang tinggi, kestabilannya dalam kompresi, dan implementasinya yang relatif sederhana. Karena kemampuannya untuk mengkonsentrasikan informasi penting dalam jumlah koefisien yang kecil, DCT sangat efektif dalam mengurangi ukuran file dengan sedikit penurunan kualitas. Oleh karena itu, DCT telah digunakan secara luas dalam berbagai format kompresi seperti JPEG (gambar), MPEG (video), dan MP3 (audio) (Najiyah, 2020).

Namun demikian, DCT juga memiliki beberapa keterbatasan. Salah satu kekurangannya adalah efek blok (*blocking artifact*), yaitu munculnya batas-batas kotak yang terlihat pada gambar hasil kompresi tinggi, terutama ketika blok-blok DCT dikompresi terlalu agresif. Untuk mengurangi efek ini, pendekatan lain seperti

transformasi *wavelet* atau algoritma hibrida sering digunakan dalam aplikasi kompresi generasi baru seperti JPEG2000.

Dalam penelitian-penelitian terkini, DCT juga mulai dikombinasikan dengan pendekatan kecerdasan buatan untuk meningkatkan performa kompresi. Misalnya, *autoencoder neural network* dapat digunakan untuk mengatur dan memfilter koefisien DCT secara adaptif berdasarkan karakteristik konten citra. Selain itu, pengembangan varian dari DCT seperti *Modified Discrete Cosine Transform* (MDCT) dan *integer* DCT juga terus dilakukan untuk memperbaiki efisiensi komputasi dan performa kualitas citra terkompresi.

2.5 Studio Code

VSCode adalah singkatan dari Visual Code Studio, yang merupakan aplikasi code editor untuk membantu proses pengembangan sebuah aplikasi. *Software* ini dikembangkan oleh perusahaan teknologi raksasa ternama, Microsoft. Meskipun begitu, VSCode mendukung untuk dioperasikan pada perangkat selain Windows, seperti Linux dan Mac OS.

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node.Js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via *marketplace Visual Studio Code* seperti : C++, C#, Python, Go, Java, PHP, dst (Nendya et al., 2023).

Fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan teks

editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code.

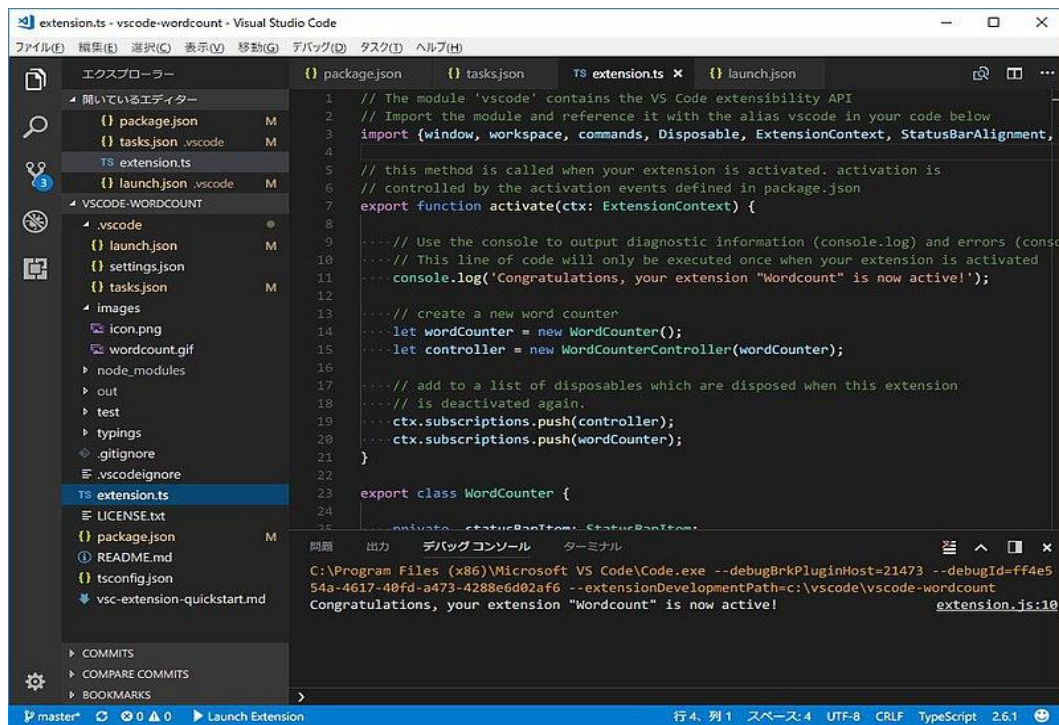
2.5.1 Pengoperasian Visual Code

Fitur utama dari Visual Code Studio mencakup berbagai alat dan fitur basic editing. Dalam fitur ini, kamu bisa menemukan berbagai *tools* seperti *Multiple Selection*, *Column Selection*, hingga *Keyboard Shortcuts*. Selain *tools* yang sudah disebutkan, VSCode juga dilengkapi dengan fitur *Auto Save* serta *Hot Exit*. Kedua fitur ini memungkinkan kamu untuk mengedit tanpa takut lupa menyimpan perubahan terakhir pada file. Pekerjaanmu di VSCode akan tersimpan otomatis tanpa perlu kamu simpan secara manual. Dengan begitu, kamu tidak perlu khawatir jika perangkat komputer atau laptop mengalami gangguan di tengah-tengah proses editing (Nendya et al., 2023).

Versi *default* VSCode mendukung fitur code debugging untuk platform dengan bahasa pemrograman Node.js. Akan tetapi bukan berarti fitur ini tidak tersedia untuk bahasa pemrograman lainnya.

Memang Visual Studio Code tergolong unggul dari segi kelengkapan fitur. Ini dikarenakan adanya peran dari *extension marketplace* yang mana memungkinkan pengguna menjadi lebih leluasa di dalam menambahkan fitur tertentu. Menariknya lagi semua itu bisa diperoleh secara gratis. Kendati Visual Studio Code mempunyai segudang fitur, nyatanya software tersebut bisa dikatakan sangatlah ringan. Aplikasi tersebut tidak memakan banyak ruang penyimpanan dan penggunaan RAM tergolong cukup ramah. Jadi kalian tidak membutuhkan perangkat dengan spesifikasi tinggi dalam menjalankan softwarenya. Keunggulan

lain visual studio code adalah mendukung cukup banyak untuk bahasa pemrograman. Apapun bahasa yang disenangi dapat dipakai disini entah itu bahasa yang memang populer ataupun jarang digunakan. Keunggulan ini tidak terlepas dari adanya fitur *extension marketplace*.



Gambar 2. 2 Visual Code

(Sumber : Nendya et al., 2023).

2.6 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*), merupakan bahasa pemrograman pada sisi server yang memperbolehkan *programmer* menyisipkan perintah-perintah perangkat lunak *web server* (*Apache*, *IIS*, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke *browser* yang me-*request*-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman *web* setiap kali tampilan tanggal dibutuhkan. Sesuai dengan fungsinya

yang berjalan di sisi *server* maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application* (Nadialista Kurniawan, 2021).

PHP telah menjadi bahasa *scripting* untuk keperluan umum yang pada awalnya hanya digunakan untuk pembangunan *web* yang menghasilkan halaman *web* dinamis. Untuk tujuan ini, kode PHP tertanam ke dalam dokumen sumber *HTML* dan diinterpretasikan oleh *server web* dengan modul PHP *prosesor*, yang menghasilkan dokumen halaman *web*. Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris-baris perintah modus dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program di channel *output* standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk *server web* yang paling modern dan sebagai penerjemah mandiri pada sebagian besar sistem operasi dan komputer *platform* (Napitu et al., 2021).

2.6.1 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah *lisensi GPL* (*general public license*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*structured query language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Aldo et al., 2021).

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya.

2.7 UML (Unified Modelling Language)

Unified modelling language merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *object oriented* (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen (Sukma & Abhyanda, 2020).

UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu *grady booch*, OOD (*object-oriented design*), *jim rumbaugh*, OMT (*object modelling technique*), dan *ivar jacobson* OOSE (*object-oriented software engineering*). UML mempunyai tiga kategori utama yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi.

Unified modelling language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. Sejarah UML sendiri terbagi dalam dua *fase*, sebelum dan sesudah munculnya UML. Dalam *fase* sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990-an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi (Noviana, 2022).

Fase kedua, dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *object management group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika *grady booch* dengan metode OOD (*object-oriented design*), *jim rumbaugh* dengan metode OMT (*object modelling technique*) mereka ini bekerja pada *rasional software corporation* dan *ivar jacobson* dengan metode OOSE (*object-oriented software engineering*) yang bekerja pada perusahaan *objectory rasional*. (Anisah R, 2017)






Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *request for proposal* (RFP) yang dikeluarkan oleh OMG.






1. *Use Case*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case*

digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.
memiliki dua istilah yaitu *system use case*; interaksi dengan sistem.

Tabel 2. 1 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen.
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi






(Sumber : Noviana 2022)

2. *Activity diagram*

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*.

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

Tabel 2. 2 Simbol Activity Diagram


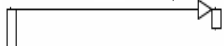

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity</i> <i>Final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

(Sumber : Noviana 2022)

3. *Sequence* diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

Tabel 2. 3 Simbol *Sequence Diagram*






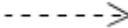
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi


(Sumber : Noviana 2022)

4. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan *name*. kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem.

Tabel 2. 4 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi

			elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Noviana 2022)