

BAB II

LANDASAN TEORI

2.1 Dokumen

Dokumen adalah kumpulan informasi tertulis atau cetak yang disimpan dalam bentuk fisik atau digital. Dokumen dapat berupa surat, laporan, memo, kontrak, atau dokumen lainnya yang berisi informasi penting yang harus disimpan dan diakses kembali di masa depan. Dokumen juga dapat berupa rekaman audio atau video yang disimpan dalam bentuk digital. Dalam era digital saat ini, dokumen sering disimpan dalam bentuk elektronik atau digital. Dokumen elektronik dapat disimpan dalam berbagai format seperti PDF, DOC, atau XLS. Dokumen digital dapat disimpan dalam berbagai media seperti hard drive, flash drive, atau cloud storage [1].

Namun, meskipun dokumen disimpan dalam bentuk digital, penting untuk memastikan keamanan dan privasi dokumen tersebut. Dokumen yang tidak diatur dengan baik dapat mudah diakses oleh pihak yang tidak berwenang dan dapat menyebabkan kerugian yang signifikan.

2.2 Gambar

Gambar merupakan salah satu bentuk representasi visual yang digunakan untuk menyampaikan informasi atau pesan [11]. Gambar dapat diartikan sebagai suatu bentuk representasi visual yang dibuat dengan menggunakan berbagai media seperti pensil, cat, atau tinta. Gambar dapat digunakan untuk berbagai keperluan seperti menggambarkan objek, memvisualisasikan konsep, atau sebagai media ekspresi seni.

Gambar juga dapat diartikan sebagai suatu bentuk komunikasi visual yang memiliki bahasa tersendiri. Bahasa visual yang digunakan dalam gambar dapat berupa garis, warna, bentuk, dan tekstur. Dalam hal ini, gambar dapat dianggap sebagai suatu bentuk bahasa yang dapat digunakan untuk menyampaikan pesan atau informasi. Selain itu, gambar juga dapat diartikan sebagai suatu bentuk representasi visual yang memiliki nilai estetika. Gambar dapat dianggap sebagai suatu bentuk seni

yang memiliki nilai estetika yang tinggi. Dalam hal ini, gambar dapat digunakan sebagai media ekspresi seni yang dapat menghasilkan karya yang indah dan memikat.

Berdasarkan definisi-definisi di atas, dapat disimpulkan bahwa gambar merupakan suatu bentuk representasi visual yang dapat digunakan untuk berbagai keperluan seperti menggambarkan objek, memvisualisasikan konsep, atau sebagai media ekspresi seni. Gambar juga memiliki bahasa tersendiri yang dapat digunakan untuk menyampaikan pesan atau informasi, serta memiliki nilai estetika yang tinggi sebagai suatu bentuk seni.

2.3 *Optical Character Recognition*

Optical Character Recognition (OCR) adalah teknologi yang digunakan untuk mengenali dan mengubah teks yang tercetak atau ditulis tangan menjadi format digital yang dapat diolah oleh komputer. OCR telah berkembang pesat sejak pertama kali diperkenalkan pada tahun 1950-an dan terus mengalami perbaikan dan peningkatan kinerja hingga saat ini. Setelah tahun 2010, OCR telah menjadi lebih canggih dan dapat mengenali teks dengan akurasi yang lebih tinggi. Teknologi OCR saat ini dapat mengenali berbagai jenis font, bahasa, dan karakter khusus. Selain itu, OCR juga dapat mengenali tulisan tangan dan mengubahnya menjadi teks digital [12].

Salah satu perkembangan terbaru dalam teknologi OCR adalah penggunaan deep learning dan neural network. Dengan menggunakan teknologi ini, OCR dapat belajar dan meningkatkan kinerjanya secara mandiri, sehingga dapat mengenali teks dengan akurasi yang lebih tinggi. Namun, meskipun teknologi OCR telah berkembang pesat, masih terdapat beberapa tantangan yang perlu diatasi. Salah satu tantangan utama adalah pengenalan teks dalam kondisi yang buruk, seperti teks yang terdistorsi atau terpotong. Selain itu, pengenalan teks dalam bahasa yang kurang umum atau karakter khusus juga masih menjadi tantangan.

Contoh penerapan teknologi OCR adalah pada pengolahan dokumen bisnis, seperti faktur, laporan keuangan, dan kontrak. Dalam pengolahan dokumen bisnis,

OCR dapat digunakan untuk mengekstrak data dari dokumen yang telah di-scan, seperti nomor faktur, tanggal, dan jumlah, yang kemudian dapat diimpor ke sistem akuntansi atau database untuk pengolahan lebih lanjut. Namun, meskipun teknologi OCR sangat berguna dalam pengolahan dokumen digital, masih ada beberapa masalah yang dihadapi, seperti kesalahan pengenalan karakter, terutama pada dokumen dengan font yang tidak standar atau teks yang rusak. Oleh karena itu, perlu dilakukan evaluasi dan perbaikan terus-menerus pada teknologi OCR untuk meningkatkan akurasi dan efektivitasnya dalam pengolahan dokumen digital [13].

Berikut adalah tahapan umum yang terlibat dalam proses OCR:

1. Pemindaian (Scanning)

Tahap pertama dalam OCR adalah memindai dokumen yang berisi teks menggunakan perangkat pemindai (scanner) atau kamera digital. Pemindaian menghasilkan gambar digital dari dokumen yang akan diolah selanjutnya.

2. Pra-Pemrosesan (Preprocessing)

Setelah pemindaian, gambar hasil pemindaian perlu diproses untuk memperbaiki kualitas dan mempersiapkannya untuk pengenalan karakter. Tahap pra-pemrosesan mencakup langkah-langkah seperti pengaturan kontras dan kecerahan, penghapusan noise, penyesuaian orientasi, penghapusan garis atau bayangan, dan pemadatan karakter.

3. Segmentasi Karakter (Character Segmentation)

Pada tahap ini, gambar yang telah diolah kemudian dipecah menjadi unit karakter individu. Proses segmentasi karakter melibatkan pengenalan dan pemisahan karakter satu per satu. Ini dapat mencakup penggunaan metode pemisahan seperti pemisahan berdasarkan spasial, analisis pemisahan garis horizontal atau vertikal, atau pengenalan bentuk dan kontur karakter.

4. Ekstraksi Fitur (Feature Extraction)

Setelah karakter dipisahkan, tahap selanjutnya adalah mengekstraksi fitur-fitur penting dari setiap karakter. Pada tahap ini, fitur-fitur seperti bentuk, ukuran, orientasi, dan tekstur karakter diekstraksi untuk membantu dalam pengenalan karakter yang akurat.

5. Pengenalan Karakter (Character Recognition)

Tahap pengenalan karakter adalah inti dari OCR. Pada tahap ini, fitur-fitur yang telah diekstraksi dibandingkan dengan basis data referensi atau model pembelajaran mesin untuk mengenali karakter yang tepat. Algoritma pengenalan karakter yang umum digunakan termasuk metode template matching, jaringan saraf tiruan, dan metode berbasis statistik seperti metode Markov tersembunyi (Hidden Markov Models).

6. Koreksi dan Verifikasi (Correction and Verification)

Setelah karakter dikenali, tahap ini melibatkan koreksi dan verifikasi hasil pengenalan karakter. Algoritma dan teknik pemrosesan teks lanjutan dapat digunakan untuk mengoreksi kesalahan pengenalan karakter yang terjadi selama proses OCR. Misalnya, metode koreksi berdasarkan kamus, model bahasa, atau pemrosesan konteks dapat digunakan untuk meningkatkan akurasi hasil pengenalan.

7. Output Teks (Text Output)

Tahap terakhir dalam OCR adalah menghasilkan output teks yang terstruktur dari dokumen yang dipindai. Setelah karakter dikenali dan dikoreksi, teks hasilnya dapat disimpan dalam berbagai format seperti file teks sederhana, file PDF yang dapat dicari, atau dalam format yang lebih kaya seperti HTML atau XML.

Akurasi Optical Character Recognition (OCR) bergantung pada beberapa faktor, termasuk:

1. Kualitas Gambar: OCR biasanya lebih akurat ketika gambar berkualitas tinggi dengan resolusi yang cukup dan kontras yang baik antara teks dan latar belakangnya.

2. Font dan Ukuran Teks: Beberapa font lebih mudah dikenali daripada yang lain. Ukuran teks yang sangat kecil atau sangat besar dapat mengurangi akurasi.
3. Kompleksitas Halaman: Halaman dengan banyak kolom, grafik, atau elemen desain lainnya dapat mengurangi akurasi OCR.
4. Bahasa: Kinerja OCR mungkin berbeda tergantung pada bahasa yang digunakan. Misalnya, OCR yang dioptimalkan untuk bahasa Inggris mungkin tidak seakurat untuk bahasa lain.
5. Software OCR: Ada banyak software OCR di pasar, dan masing-masing memiliki kelebihan dan kekurangan sendiri.

Secara umum, perangkat lunak OCR terbaik di pasar dapat mencapai tingkat akurasi hingga 98-99% pada dokumen dengan kualitas baik dan format sederhana. Namun, di kondisi yang kurang ideal, akurasi bisa jauh lebih rendah.

2.4 *Image Processing*

Pengolahan citra atau *Image Processing* adalah suatu sistem dimana proses dilakukan dengan masukan (input) berupa citra (*Image*) dan hasilnya (output) juga berupa citra (*Image*). Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka *Image processing* tidak dapat dilepaskan dengan bidang computer vision [14].

Tujuan utama dari pengolahan citra adalah untuk mengubah citra menjadi bentuk digital dan melakukan operasi dengan algoritma tertentu untuk mendapatkan model atau mengekstrak informasi yang berguna pada citra.

Tahapan untuk menghapus latar belakang pada gambar dapat bervariasi tergantung pada metode yang digunakan. Berikut ini adalah beberapa tahapan umum yang terlibat dalam proses penghapusan latar belakang pada gambar:

1. Segmentasi Objek

Tahap pertama dalam penghapusan latar belakang adalah mengidentifikasi objek yang ingin dipisahkan dari latar belakang. Ada beberapa metode yang dapat digunakan untuk segmentasi objek, termasuk metode berbasis warna, deteksi tepi, atau penggunaan algoritma pemisahan seperti algoritma GrabCut atau algoritma pemisahan berdasarkan pelabelan piksel (pixel labeling).

2. Pembuatan Masker

Setelah objek tersegmentasi, langkah selanjutnya adalah membuat masker yang memisahkan objek dari latar belakang. Masker ini berisi informasi biner (misalnya, piksel putih dan hitam) yang menandakan area objek dan area latar belakang. Metode yang umum digunakan untuk membuat masker adalah dengan menggunakan teknik thresholding atau teknik segmentasi seperti algoritma GrabCut.

3. Penyempurnaan Masker

Kadang-kadang, masker yang dihasilkan mungkin tidak sempurna atau terdapat bagian dari objek yang tergabung dalam latar belakang. Tahap ini melibatkan penyempurnaan masker dengan menggunakan teknik pemrosesan citra seperti operasi morfologi (erosi dan dilasi) atau pemisahan berbasis kontur untuk menghapus noise atau memperbaiki kesalahan dalam masker.

4. Pemotongan dan Penyesuaian

Setelah masker objek terbentuk, objek dapat dipotong dari gambar asli menggunakan masker sebagai panduan. Pemotongan objek melibatkan operasi matematis piksel-per-piksel yang melibatkan penggabungan gambar objek dengan latar belakang yang baru atau latar belakang transparan.

5. Finishing dan Pembersihan

Setelah objek dipotong dari latar belakang, seringkali diperlukan tahap pembersihan dan finishing untuk menghilangkan sisa-sisa latar belakang yang mungkin masih ada atau untuk menyempurnakan hasil akhir. Ini dapat melibatkan teknik pemulusan tepi, penyesuaian warna, atau pemrosesan lanjutan lainnya untuk mencapai tampilan yang lebih alami dan halus.

Tahapan-tahapan ini memberikan gambaran umum tentang proses penghapusan latar belakang pada gambar. Penting untuk dicatat bahwa implementasinya dapat bervariasi tergantung pada algoritma dan metode yang digunakan, serta kompleksitas objek dan latar belakang yang ada dalam gambar yang akan diolah.

2.5 Website

Situs Web atau Website adalah sekumpulan halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti Internet, ataupun jaringan area lokal (LAN) melalui alamat Internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di Internet disebut pula sebagai *World Wide Web* atau lebih dikenal dengan singkatan WWW. Meskipun setidaknya halaman beranda situs Internet umumnya dapat diakses publik secara bebas, pada praktiknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa situs web mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota, atau bahkan meminta pembayaran untuk dapat menjadi anggota untuk dapat mengakses isi yang terdapat dalam situs web tersebut, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surel (*e-mail*), dan lain-lain. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi, atau karena tujuan komersial tertentu [15].

Sebuah halaman web adalah berkas yang ditulis sebagai berkas teks biasa (plain text) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML atau XHTML, kadang-kadang pula disisipi dengan sekelumit bahasa skrip. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer. Halaman-halaman web tersebut diakses oleh pengguna melalui protokol komunikasi jaringan yang disebut sebagai HTTP, sebagai tambahan untuk meningkatkan aspek keamanan dan aspek privasi yang lebih baik, situs web dapat pula mengimplementasikan mekanisme pengaksesan melalui protokol HTTPS.

2.6 *Text-to-Speech*

Text to speech (TTS) adalah teknologi yang memungkinkan konversi teks menjadi suara atau ucapan manusia dengan menggunakan program atau algoritma tertentu. Dalam sistem TTS, teks yang ditulis oleh pengguna akan diproses oleh perangkat lunak atau mesin, kemudian diubah menjadi suara manusia yang bisa didengar oleh pengguna melalui speaker atau headphone. Teknologi TTS telah banyak digunakan dalam berbagai aplikasi, seperti sistem navigasi, assistive technology untuk penyandang disabilitas, pembaca buku audio, dan lain sebagainya [16].

Berdasarkan definisi tersebut, maka suatu sistem Interactive Voice Response (IVR) yang banyak digunakan untuk layanan informasi otomatis melalui telpon (call center) tidak dapat dikategorikan sebagai sistem TTS karena hanya dapat mengucapkan kata-kata atau kalimat dengan jumlah atau kombinasi yang sangat terbatas.

Tahapan dalam konversi teks ke ucapan (Text-to-Speech, TTS) untuk membacakan dokumen melibatkan beberapa langkah berikut:

1. Analisis Teks

Tahap pertama adalah menganalisis teks dalam dokumen yang akan dibacakan. Ini melibatkan pemrosesan dan pemahaman struktur teks, seperti mengenali

paragraf, kalimat, dan frasa dalam dokumen. Analisis ini membantu dalam memberikan penekanan yang tepat dan intonasi yang sesuai saat teks diucapkan.

2. Normalisasi Teks

Tahap selanjutnya adalah melakukan normalisasi teks, yang melibatkan penyesuaian dan standarisasi teks untuk memastikan ucapan yang konsisten dan tepat. Ini dapat melibatkan penggantian kata singkatan dengan kata lengkap, konversi angka menjadi format yang sesuai untuk pengucapan, dan penyesuaian tata bahasa seperti konjugasi kata kerja.

3. Segmentasi Teks

Setelah normalisasi, teks dalam dokumen akan dipisahkan menjadi unit pembacaan yang lebih kecil, seperti kalimat atau frasa. Ini membantu dalam memberikan jeda yang tepat dan pengaturan intonasi yang sesuai dalam output TTS.

4. Konversi ke Fonem

Tahap berikutnya adalah mengonversi teks menjadi fonem, yaitu unit bunyi dalam bahasa yang digunakan untuk mengucapkan kata-kata. Setiap kata dalam dokumen akan dianalisis secara fonetik dan dipetakan ke urutan fonem yang sesuai. Proses ini membantu dalam menghasilkan pengucapan yang akurat dan alami.

5. Pengaturan Ucapan

Setelah fonem diperoleh, TTS akan menggunakan aturan fonetik, sintaksis, dan intonasi untuk menghasilkan pengaturan ucapan yang sesuai. Ini melibatkan menentukan durasi, stres, intonasi, dan pengaturan lainnya untuk menghasilkan output TTS yang lebih alami dan dapat dimengerti.

6. Sintesis Ucapan

Tahap kunci dalam TTS adalah sintesis ucapan, di mana fonem dan pengaturan ucapan diterjemahkan menjadi suara manusia yang dapat didengar. Metode sintesis dapat berbeda, termasuk sintesis konkatif (*concatenative synthesis*), sintesis aturan (*rule-based synthesis*), atau sintesis berbasis unit (*unit-based synthesis*). Proses ini

melibatkan penggunaan database suara manusia atau penggunaan model pembelajaran mesin untuk menghasilkan suara ucapan.

7. Output Ucapan

Tahap terakhir adalah menghasilkan output TTS dalam bentuk suara yang dapat didengar. Output ini dapat disimpan sebagai file audio, ditransmisikan melalui speaker, atau digunakan dalam aplikasi atau perangkat yang memerlukan pembacaan teks.

2.7 Framework

Framework adalah sebuah kerangka kerja atau struktur yang digunakan untuk membangun dan mengembangkan aplikasi atau software. Framework berisi serangkaian aturan, konvensi, dan prosedur yang telah dirancang sebelumnya untuk membantu pengembang membangun aplikasi dengan lebih cepat, efisien, dan konsisten. Framework biasanya terdiri dari kumpulan kode, modul, dan library yang dapat digunakan oleh pengembang untuk mempermudah pengembangan aplikasi. Selain itu, framework juga membantu dalam memecahkan masalah yang kompleks dan mengurangi risiko kesalahan dalam pengembangan aplikasi [17]. Beberapa contoh framework yang populer di dunia pengembangan software adalah Ruby on Rails, Laravel, AngularJS, CodeIgniter dan ReactJS.

2.8 PHP

PHP (PHP: Hypertext Preprocessor) adalah bahasa pemrograman opensource yang digunakan untuk membangun aplikasi web dinamis. PHP digunakan secara luas di seluruh dunia dan merupakan salah satu bahasa pemrograman yang paling populer untuk membangun situs web dan aplikasi web. PHP memungkinkan pengembang untuk membuat halaman web yang dinamis dan interaktif dengan menghasilkan konten dinamis dan memproses formulir atau input dari pengguna [18].

PHP berjalan pada sisi server, yang berarti bahwa kode PHP dieksekusi pada server web sebelum halaman web dihasilkan dan dikirim ke browser pengguna. PHP dapat berintegrasi dengan server web seperti Apache, Nginx, dan IIS, serta

berinteraksi dengan database seperti MySQL, PostgreSQL, dan Oracle. PHP memiliki banyak fitur yang memungkinkan pengembang untuk membuat aplikasi web yang kuat dan fleksibel, seperti dukungan untuk pemrograman berorientasi objek, pengelolaan memori yang otomatis, dan kemampuan untuk berinteraksi dengan berbagai jenis database. Selain itu, PHP juga dapat diintegrasikan dengan banyak kerangka kerja (framework) seperti Laravel, Symfony, dan CodeIgniter yang mempermudah dalam pengembangan aplikasi web.

2.9 Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk membangun aplikasi web interaktif. JavaScript dikembangkan oleh Netscape pada tahun 1995 dan sekarang menjadi salah satu bahasa pemrograman paling populer di dunia. JavaScript berjalan pada sisi klien, yang berarti bahwa kode JavaScript dieksekusi pada browser pengguna setelah halaman web telah diunduh. Hal ini memungkinkan JavaScript untuk membuat halaman web lebih interaktif dengan menangani interaksi pengguna seperti klik tombol, pengisian formulir, dan animasi [19].

JavaScript memiliki banyak fitur yang memungkinkan pengembang untuk membuat aplikasi web yang kuat dan dinamis, seperti dukungan untuk pemrograman berorientasi objek, manipulasi DOM (Document Object Model), dan interaksi dengan server web melalui teknologi AJAX. JavaScript juga dapat diintegrasikan dengan berbagai kerangka kerja (framework) seperti React, Angular, dan Vue, yang mempermudah dalam pengembangan aplikasi web dengan menggunakan konsep-konsep pemrograman modern.

2.10 CSS

CSS (Cascading Style Sheets) adalah bahasa pemrograman yang digunakan untuk mengatur tampilan (styling) halaman web. CSS memungkinkan pengembang untuk memisahkan antara struktur halaman web (HTML) dengan tampilannya, sehingga membuat desain dan perubahan tampilan lebih mudah dan efisien. CSS digunakan untuk mengatur berbagai properti tampilan seperti warna, ukuran font, layout halaman, dan animasi. CSS juga memungkinkan pengembang untuk membuat tampilan halaman web yang responsif dan terlihat bagus pada berbagai

perangkat dan ukuran layar yang berbeda [20]. CSS juga dapat diintegrasikan dengan berbagai kerangka kerja (framework) seperti Bootstrap, Foundation, dan Bulma, yang menyediakan komponen dan gaya yang telah dibuat sebelumnya untuk mempercepat proses pengembangan.

2.11 HTML

HTML (Hypertext Markup Language) adalah bahasa markup yang digunakan untuk membuat halaman web. HTML memungkinkan pengembang untuk membuat struktur dan konten halaman web, seperti teks, gambar, video, dan tautan. HTML terdiri dari berbagai tag (tag HTML), yang digunakan untuk menentukan elemen halaman web dan strukturnya. Setiap tag HTML memiliki fungsi dan arti yang berbeda, dan dapat dikombinasikan dengan tag lain untuk membuat halaman web yang kompleks. HTML merupakan salah satu teknologi inti dari World Wide Web (WWW) dan digunakan di seluruh dunia untuk membuat dan mengakses halaman web. HTML selalu dikombinasikan dengan CSS dan JavaScript untuk menghasilkan halaman web yang dinamis dan interaktif [21].

2.12 Application Programming Interface

API (Application Programming Interface) adalah kumpulan protokol, aturan, dan alat pemrograman yang digunakan untuk mengembangkan perangkat lunak dan aplikasi. API menyediakan cara bagi pengembang untuk berkomunikasi dan mengintegrasikan aplikasi dengan sistem atau layanan lain. API biasanya digunakan untuk berinteraksi dengan layanan web, seperti Facebook, Twitter, Google Maps, dan banyak lagi. Dengan menggunakan API, pengembang dapat mengambil data atau menggunakan layanan yang disediakan oleh layanan web tersebut dan mengintegrasikannya ke dalam aplikasi mereka sendiri. API juga dapat membantu mengurangi kerumitan pengembangan perangkat lunak dengan memungkinkan pengembang untuk mengakses dan menggunakan fungsi atau layanan tertentu dari sistem atau aplikasi lain tanpa harus memahami seluruh detail internal sistem atau aplikasi tersebut [22].

2.12.1 OCR API

OCR API adalah layanan web yang memungkinkan pengguna untuk mengirimkan gambar atau dokumen dalam format digital dan mendapatkan hasilnya dalam bentuk teks yang dapat diedit. OCR API menggunakan teknologi *Optical Character Recognition* (OCR) untuk mengekstrak teks dari gambar atau dokumen digital. Dengan menggunakan OCR API, pengguna dapat menghemat waktu dan usaha dalam menyetik ulang dokumen atau memasukkan data dari gambar ke dalam program atau aplikasi. OCR API juga dapat membantu meningkatkan akurasi dan efisiensi dalam pengolahan data dan informasi.

2.12.2 Remove.bg API

Remove.bg API adalah layanan web yang memungkinkan pengguna untuk menghapus latar belakang gambar secara otomatis dan cepat. Remove.bg API menggunakan teknologi pengenalan objek dan kecerdasan buatan (AI) untuk secara otomatis mengenali objek utama pada gambar dan menghapus latar belakangnya. Dengan menggunakan Remove.bg API, pengguna dapat menghilangkan latar belakang gambar dalam hitungan detik tanpa perlu memiliki keahlian khusus dalam pengeditan gambar. Remove.bg API juga dapat membantu meningkatkan kualitas gambar, meningkatkan efisiensi dalam produksi konten, dan mempercepat proses desain grafis.

2.12.3 DocTranslator API

API DocTranslator adalah API versi beta yang memungkinkan Anda menerjemahkan dokumen dalam berbagai format, termasuk Word, PDF, Excel, PowerPoint, OpenOffice, dan teks. API ini menggunakan layanan Google Translate untuk menerjemahkan dokumen Anda, dan menjaga tata letak asli dokumen.

2.13 Visual Studio Code

Visual Studio Code adalah editor kode sumber yang dikembangkan oleh Microsoft. Visual Studio Code adalah aplikasi open-source dan dapat digunakan pada berbagai sistem operasi, termasuk Windows, macOS, dan Linux. Visual Studio Code menawarkan berbagai fitur seperti tampilan kode sumber yang mudah dibaca, kemampuan debugging yang kuat, dan integrasi dengan berbagai bahasa

pemrograman, termasuk JavaScript, TypeScript, Python, dan banyak lagi. Visual Studio Code juga dapat diperluas melalui plugin dan ekstensi, yang memungkinkan pengguna untuk menambahkan fitur dan fungsionalitas baru ke editor [23].

2.14 Webservice

Web service merupakan aplikasi yang berisi sekumpulan basis data (database) dan perangkat lunak (software) atau bagian dari program perangkat lunak yang diakses secara remote oleh piranti dengan perantara tertentu. Melalui web service, memungkinkan pengguna untuk mengatasi permasalahan berupa interoperability dan mengintegrasikan sistem berbeda [24].

Pada umumnya, web service memiliki ciri khusus berupa URL layaknya web. Yang membuat berbeda adalah interaksi yang diberikan oleh web service itu sendiri. URL pada web service hanya mengandung sekumpulan informasi, perintah, dan konfigurasi (sintaks yang berguna untuk membangun fungsi tertentu dari aplikasi). Web service mampu menukar data tanpa memandang sumber database, bahasa yang digunakan, dan pada platform apa data tersebut dikonsumsi. Kemampuan itulah yang memungkinkan web service menjadi jembatan penghubung untuk berbagai sistem [25].

2.15 Unified Modelling Language

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh Object Management Group dengan versi awal 1.0 pada bulan Januari 1997. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software. UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor scalability, robustness, security, dan sebagainya [26].

2.15.1 Use Case Diagram

Use case diagram adalah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya.

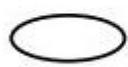
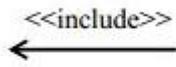
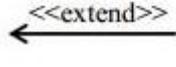
Tentunya, use case diagram merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan, tentu perlunya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi sistem itu sendiri, seperti yang terdapat pada use case diagram.

Adapun, fungsi dari use case diagram sebagai berikut:

1. Berguna memperlihatkan proses aktivitas secara urut dalam sistem.
2. Mampu menggambarkan proses bisnis, bahkan menampilkan urutan aktivitas pada sebuah proses.
3. Sebagai bridge atau jembatan antara pembuat dengan konsumen untuk mendeskripsikan sebuah sistem.

Berikut Tabel simbol-simbol yang terdapat didalam use case diagram.

Tabel 2. 1 Daftar Simbol Use Case Diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan use case
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

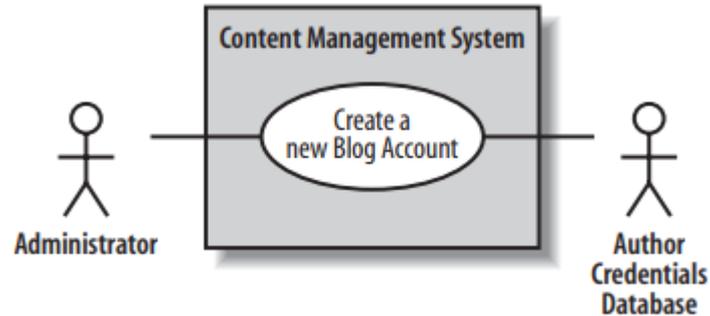
Sumber : <https://www.dicoding.com/blog/contoh-use-case-diagram/>

Komponen – komponen pada Use Case Diagram [27] :

1. Sistem

Sebuah sistem digambarkan ke dalam bentuk persegi. Fungsinya untuk membatasi use case dengan interaksi dari luar sistem. Sistem pada umumnya

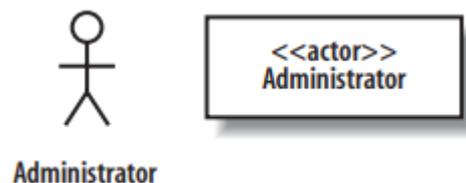
diberikan label yang sesuai. Namun, umumnya sistem ini tidaklah diberi gambar karena kita tidak terlalu memberikan makna pada sebuah diagram .



Gambar 2. 1 System Boundaries

2. Actor

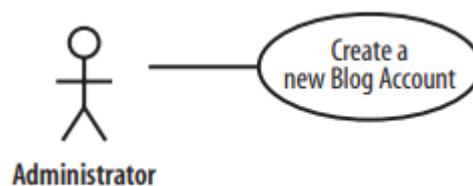
Banyak yang berspekulasi bahwa actor adalah bagian dari diagram. Padahal apabila kita mencari informasi lebih dalam mengenai soal ini, ternyata actor bukanlah bagian dari diagram. Peran actor sangat penting, tentunya menciptakan use case jadi lebih mudah. Fungsi Actor menjelaskan siapa yang berinteraksi dengan sistem. Actor akan memberikan informasi kepada sistem, serta menerima informasi dari sistem. Keduanya bisa terjadi secara bersamaan. Actor tidak memberikan kontrol terhadap sistem, namun hanya memberikan gambaran mengenai hubungannya dengan sistem.



Gambar 2. 2 Actor use case

3. Use Case

Use case adalah komponen gambaran fungsional dalam sebuah sistem. Sehingga konsumen maupun pembuat saling mengenal dan mengerti mengenai alur sistem yang akan dibuat.



Gambar 2. 3 Use Case

2.15.2 Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. Activity diagram merupakan pengembangan dari Use Case yang memiliki alur aktivitas. Activity diagram mesti digunakan sejajar (horizontal) dengan teknik pemodelan lainnya, seperti diagram Use Case dan diagram State. Kamu bisa menggunakan activity diagram agar dapat memodelkan alur kerja sistem dengan baik. Activity diagram berfungsi juga untuk menganalisis diagram use case dengan cara mendeskripsikan aktor, tindakan yang perlu dilakukan, dan kapan harus terjadi [26].

Berikut merupakan komponen yang terdapat di dalam activity diagram.

Tabel 2. 2 Daftar Simbol Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

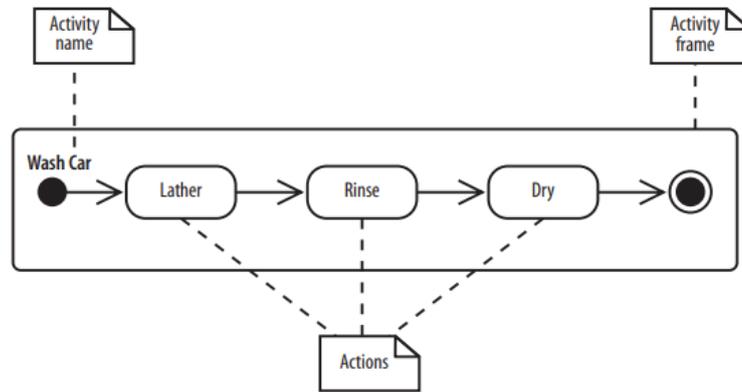
Komponen – komponen yang terdapat pada activity diagram [27] :

1. Start Point atau Initial State (Titik Mulai/Status Mulai)

Start Point adalah lingkaran hitam kecil. Biasanya digunakan untuk menandakan status awal, tindakan awal, atau titik awal aktivitas untuk setiap activity diagram.

2. Activity (Aktivitas)

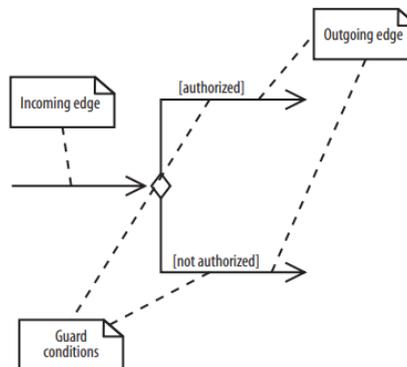
Activity merupakan aktivitas yang dilakukan atau sedang terjadi dalam sistem. Biasanya diawali dengan “kata kerja” dari aktivitas yang dilakukan.



Gambar 2. 4 Activity and Actions

3. Decision atau Percabangan

Percabangan atau decision merupakan suatu titik atau point yang mengindikasikan suatu kondisi di mana adanya kemungkinan dalam perbedaan transisi. Hal tersebut diperlukan ketika sistem yang dimiliki memiliki beberapa kemungkinan atau jalan alternatif.



Gambar 2. 5 Decision

4. Synchronization

Synchronization dibagi menjadi 2 bagian, yaitu fork dan join.

5. Fork (percabangan) digunakan untuk memecah behaviour (tingkah laku) menjadi activity atau action (aksi) secara paralel.
6. Join (penggabungan) digunakan untuk menghubungkan kembali activity dengan action secara paralel.

7. Merge

Menggabungkan flow yang sudah dipecah menjadi beberapa bagian oleh suatu flow.

8. Swimlanes

Memecah activity diagram menjadi kolom dan baris untuk membagi tanggung jawab objek-objek yang melakukan suatu aktivitas.

9. Transition

Digunakan untuk menunjukkan aktivitas selanjutnya dan sebelumnya.

10. Notasi Akhir

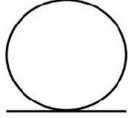
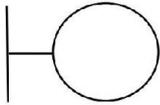
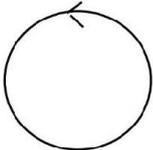
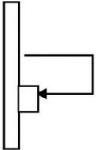
Notasi akhir digunakan untuk menandakan proses tersebut berakhir. Pada UML, notasi akhir dapat digambarkan dengan simbol sebuah bull's eye (mata sapi).

2.15.3 Sequence Diagram

Sequence diagram digunakan untuk memodelkan interaksi antara objek-objek dalam sistem perangkat lunak. Sequence diagram menunjukkan pesan-pesan atau panggilan-panggilan yang terjadi antara objek-objek dan urutan dari pesan-pesan atau panggilan-panggilan tersebut [26].

Berikut merupakan komponen yang terdapat di dalam sequence diagram.

Tabel 2. 3 Daftar Simbol Sequence Diagram

Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

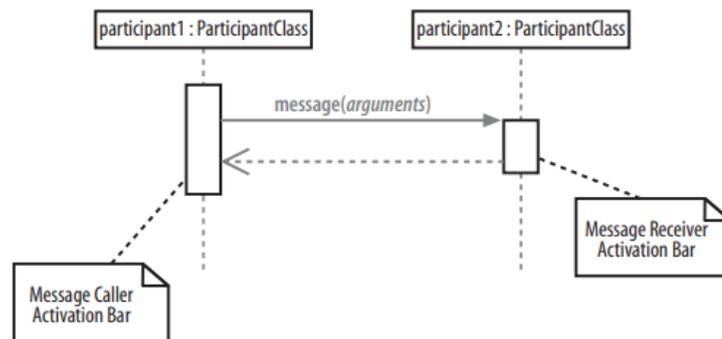
Komponen – komponen yang terdapat pada sequence diagram [27] :

1. Aktor

Komponen yang pertama adalah aktor. Komponen ini menggambarkan seorang pengguna (user) yang berada di luar sistem dan sedang berinteraksi dengan sistem. Dalam sequence diagram, aktor biasanya digambarkan dengan simbol stick figure.

2. Activation Box

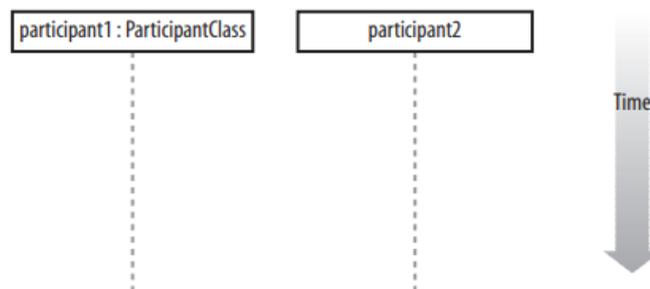
Selanjutnya ada activation box. Komponen activation box ini merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya. Semakin lama waktu yang diperlukan, maka secara otomatis activation boxnya juga akan menjadi lebih panjang. Komponen ini digambarkan dengan bentuk persegi panjang.



Gambar 2. 6 Activation

3. Lifeline

Berikutnya adalah lifeline. Komponen ini digambarkan dengan bentuk garis putus-putus. Lifeline ini biasanya memiliki kotak yang berisi objek yang memiliki fungsi untuk menggambarkan aktifitas dari objek.



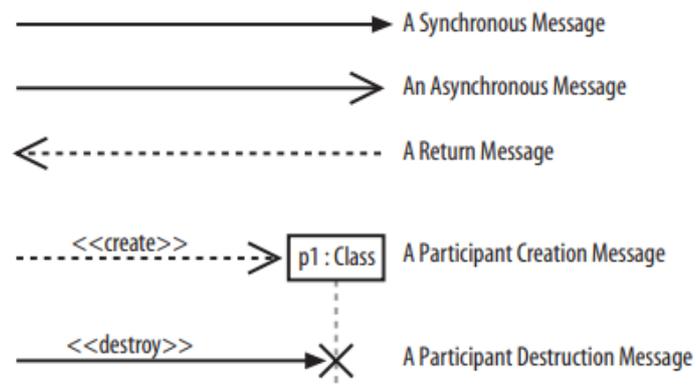
Gambar 2. 7 Lifeline

4. Objek

Komponen berikutnya adalah objek. Komponen objek ini digambarkan memiliki bentuk kotak yang berisikan nama dari objek dengan garis bawah. Biasanya objek berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem.

5. Messages

Terakhir ada messages atau pesan. Komponen ini untuk menggambarkan komunikasi antar objek. Messages biasanya muncul secara berurutan pada lifeline. Komponen messages ini direpresentasikan dengan anak panah. Inti dari sebuah diagram urutan terdapat pada komponen lifeline dan messages ini.



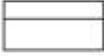
Gambar 2. 8 Messages

2.15.4 Class Diagram

Class diagram digunakan untuk memodelkan struktur data dalam sistem perangkat lunak. Class diagram menunjukkan kelas-kelas objek yang digunakan dalam sistem dan hubungan antara kelas-kelas tersebut [26].

Berikut komponen yang terdapat di dalam class diagram.

Tabel 2. 4 Daftar Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Kelas pada struktur sistem
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas
6		<i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity