

**RANCANG BANGUN *PROGRESSIVE WEB*
APPLICATION PEMBACA DAN PENGHAPUS
METADATA CITRA DIGITAL**

SKRIPSI



MUH. ADRIAN DWI PUTRA

H071191011

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
JANUARI 2024**

**RANCANG BANGUN *PROGRESSIVE WEB*
APPLICATION PEMBACA DAN PENGHAPUS
METADATA CITRA DIGITAL**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi Sistem Informasi Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

MUH. ADRIAN DWI PUTRA

H071191011

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
JANUARI 2024**

LEMBAR PERNYATAAN KEOTENTIKAN

Yang bertanda tangan di bawah ini:

Nama : Muh. Adrian Dwi Putra

NIM : H071191011

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan bahwa skripsi saya yang berjudul:

**RANCANG BANGUN *PROGRESSIVE WEB APPLICATION* PEMBACA
DAN PENGHAPUS METADATA CITRA DIGITAL**

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 4 Januari 2024

menyatakan,



Muh. Adrian Dwi Putra

NIM H071191011

**RANCANG BANGUN *PROGRESSIVE WEB APPLICATION*
PEMBACA DAN PENGHAPUS METADATA CITRA DIGITAL**

Disusun dan diajukan oleh

Muh. Adrian Dwi Putra

H071191011


Disetujui oleh:

Pembimbing Utama

Pembimbing Pendamping

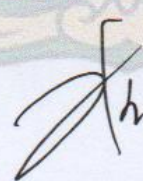

Edy Saputra Rusdi, S.Si., M.Si

NIP. 199104102020053001


A. Muh. Amil Siddik, S.Si., M.Si

NIP. 199110032019031015

Kepala Program Studi


Dr. Hendra, S.Si., M.Kom.

NIP. 197601022002121001



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Muh. Adrian Dwi Putra
NIM : H071191011
Program Studi : Sistem Informasi
Judul Skripsi : Rancang Bangun *Progressive Web Application*
Pembaca dan Penghapus Metadata Citra
Digital

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda Tangan
Ketua	: Edy Saputra Rusdi, S.Si., M.Si	(.....)
Sekretaris	: A. Muh. Amil Siddik, S.Si., M.Si	(.....)
Anggota	: Dr. Hendra, S.Si., M.Kom.	(.....)
Anggota	: Muhammad Sadno, S.Si., M.Si.	(.....)

Ditetapkan di : Makassar

Tanggal : 4 Januari 2024

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "**Rancang Bangun *Progressive Web Application* Pembaca dan Penghapus Metadata Citra Digital**". Penulisan skripsi ini bertujuan untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan Strata Satu (S1) dan memperoleh gelar Sarjana Komputer di Universitas Hasanuddin. Salam dan shalawat penulis curahkan kepada Rasulullah SAW karena telah membawa manusia ke zaman yang terang ini.


Tentu saja penulis sadar bahwa skripsi ini berhasil ditulis juga karena bantuan dari berbagai pihak. Oleh karena itu, penulis dengan tulus mengucapkan terima kasih kepada ibunda penulis dan keluarga penulis yang senantiasa ada untuk mendukung penulis baik saat waktu menulis skripsi ini maupun pada masa yang akan datang. Selain itu, dengan segala kerendahan hati penulis mengucapkan banyak-banyak terima kasih kepada (tanpa urutan tertentu):

1. Bapak **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.** selaku rektor Universitas Hasanuddin Makassar.
2. Bapak **Dr. Eng Amiruddin, M.Si.** selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta staf-staf yang telah membantu dan mengarahkan penulis dalam berbagai hal dalam urusan akademik maupun administrasi.
3. Bapak **Prof. Dr. Nurdin, S.Si., M.Si.**, selaku Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.
4. Bapak **Dr. Hendra, S.Si., M.Kom.** selaku Ketua Program Studi Sistem Informasi Universitas Hasanuddin sekaligus menjadi dosen penguji pertama penulis yang telah memberikan masukan-masukannya untuk menyempurnakan skripsi ini.
5. Bapak **Edy Saputra Rusdi, S.Si., M.Si** sebagai dosen pembimbing utama penulis yang senantiasa meluangkan waktunya untuk membimbing menulis saat menyusun skripsi ini.

6. Bapak **A. Muh. Amil Siddik, S.Si., M.Si** sebagai penasihat akademik sekaligus dosen pembimbing pendamping penulis yang senantiasa meluangkan waktunya untuk membimbing menulis saat menyusun skripsi ini.
7. Bapak **Muhammad Sadno, S.Si., M.Si.** sebagai dosen penguji kedua penulis yang telah memberikan masukan-masukannya untuk menyempurnakan skripsi ini.
8. Seluruh **Dosen-dosen Departemen Matematika** terutama **Dosen-dosen Program Studi Sistem Informasi** atas semua ilmu yang telah diajarkan ke penulis semasa perkuliahan.
9. **Sahabat-sahabat Program Studi Sistem Informasi angkatan 2019** yang telah membantu penulis semasa perkuliahan.
10. **Sahabat-sahabat internet** yang ada di Facebook penulis terutama pada grup **Lily MS** atas dukungan yang mereka berikan.
11. Komunitas **LÖVE Game Framework** dan grup **SB Off-Topic School Idol Festival** di Discord yang telah memberi penulis umpan balik awal (*early feedback*) mengenai aplikasi yang penulis rancang dan bangun untuk menyelesaikan skripsi ini.
12. **Seluruh responden** yang telah meluangkan waktunya untuk mengisi kuesioner yang penulis berikan demi keberhasilan skripsi ini.

Penulis sadar bahwa skripsi ini jauh dari kata sempurna, tetapi penulis tetap berusaha agar skripsi ini sesempurna mungkin. Maka dari itu penulis mengharapkan kritik dan saran yang membangun untuk menyempurnakan skripsi ini dan kedepannya.

Makassar, 4 Januari 2024


Muh. Adrian Dwi Putra

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan dibawah ini:

Nama : Muh. Adrian Dwi Putra
NIM : H071191011
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

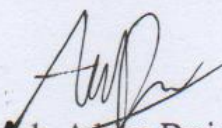
Rancang Bangun *Progressive Web Application* Pembaca dan Penghapus Metadata Citra Digital

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada Tanggal 4 Januari 2024

Yang Menyatakan


(Muh. Adrian Dwi Putra)

ABSTRAK

Perkembangan dunia digital sekarang membolehkan pengguna mengunggah citra digital ke media sosial mereka. Media sosial tersebut harus bisa melindungi privasi penggunanya dengan menghapus metadata citra penggunanya. Metadata citra adalah data sensitif yang tersembunyi pada berkas citra. Sayangnya, aplikasi penghapus metadata yang sudah ada kurang bisa digunakan oleh pengguna awam. *Progressive Web Application* (PWA) adalah aplikasi web yang dapat diinstal sebagai aplikasi terpisah dan dapat berjalan meski *offline*. Maka dari itu dirancang dan dibangun aplikasi web pembaca dan penghapus metadata citra digital menggunakan HTML, CSS, dan JavaScript yang kemudian dibuat menjadi PWA. Metode *eXtreme Programming* digunakan agar aplikasi dapat dirancang dan dibangun dengan cepat dan berulang. Hasilnya adalah, pada pengujian Lighthouse, Lighthouse memberi skor rata-rata 94,75 yang berarti aplikasi sudah sangat baik. Kemudian pada hasil *acceptance testing*, aplikasi memiliki skor *acceptance* sebesar 90,21% yang berarti aplikasi sudah sangat diterima oleh pengguna.

Kata kunci: citra digital, metadata, pwa

ABSTRACT

The development of digital world now allows users to upload digital images to their social media. Those social media should protect the privacy of their users by deleting the metadata inside their user digital images. Image metadata is a sensitive data hidden inside an image file. Unfortunately, existing metadata removal application are not easy to use by ordinary users. Progressive Web Application (PWA) is a web application that can be installed standalone and works even while offline. Therefore, a web application was designed and built to read and remove image metadata using HTML, CSS, and JavaScript which then made into a PWA. *eXtreme Programming* methodology is used to develop the application so that it can be designed quickly and iteratively. The results are, in Lighthouse test, Lighthouse gave an average score of 94.75, which means the application is very good. Then in the acceptance testing, the application has an acceptance score of 90.21%, which means the application has been highly accepted by users.

Keyword: digital image, metadata, pwa

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN KEOTENTIKAN	ii
HALAMAN PERSETUJUAN PEMBIMBING	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	v
PERSETUJUAN PUBLIKASI TUGAS AKHIR	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian	2
BAB II TINJAUAN PUSTAKA.....	4
2.1. Berkas Biner.....	4
2.2. Jenis Data Biner	4
2.3. Citra Digital.....	5
2.4. Data Pribadi.....	6
2.5. Berkas Citra JPEG.....	6
2.6. <i>Exchange Image File Format</i>	7
2.7. <i>Tag Image File Format</i>	8
2.8. <i>Comma-Separated Values</i>	10
2.9. HTML, CSS, dan JavaScript.....	11
2.10. <i>Progressive Web Application</i>	11
2.11. <i>Service Worker</i>	12
2.12. Metodologi Agile.....	13
2.13. <i>Extreme Programming</i>	14
2.14. <i>Flowchart</i>	15
2.15. <i>Unified Modeling Language</i>	17

2.16. Pengujian <i>Black Box</i>	19
2.17. Pengujian Lighthouse.....	20
2.18. <i>User Acceptance Testing</i>	22
2.19. Penelitian Terkait	23
BAB III METODE PENELITIAN	25
3.1. Waktu dan Lokasi Penelitian	25
3.2. Alur Penelitian	26
3.3. Rancangan Sistem	28
3.4. Instrumen Penelitian.....	32
BAB IV HASIL PENELITIAN.....	35
4.1. Implementasi Sistem	35
4.2. Antarmuka Sistem.....	40
4.3. Pengujian <i>Black Box</i>	43
4.4. Pengujian Lighthouse.....	47
4.5. <i>Acceptance Testing</i>	48
BAB V PENUTUP	53
5.1. Kesimpulan	53
5.2. Saran.....	53
DAFTAR PUSTAKA	55

DAFTAR GAMBAR

Gambar 2.1 Perbandingan kemampuan dan jangkauan jenis-jenis aplikasi.....	12
Gambar 2.2 Posisi <i>Service Worker</i> pada aplikasi web	13
Gambar 2.3 Siklus metodologi <i>agile</i>	14
Gambar 2.4 Tahapan perencanaan sampai masukan pada metodologi XP	15
Gambar 3.1 Alur penelitian.	26
Gambar 3.2 Kerangka tampilan awal.	28
Gambar 3.3 <i>Activity diagram</i> pengguna memilih berkas citra	29
Gambar 3.4 Kerangka tampilan setelah pengguna memilih citra.....	29
Gambar 3.5 <i>Activity diagram</i> pengguna melihat dan mengekspor metadata	30
Gambar 3.6 <i>Activity diagram</i> pemilihan metadata yang akan dihapus	30
Gambar 3.7 <i>Activity diagram</i> hapus metadata dan simpan citra baru.	31
Gambar 3.8 Interaksi pengguna dengan sistem sebagai <i>use-case diagram</i>	32
Gambar 4.1 Diagram alir (<i>flowchart</i>) pembaca metadata	36
Gambar 4.2 Diagram alir penghapusan metadata.....	37
Gambar 4.3 Diagram alir fitur <i>export to CSV</i>	38
Gambar 4.4 Kode QR untuk membuka kode sumber implementasi sistem	39
Gambar 4.5 Tampilan utama	40
Gambar 4.6 Tampilan daftar metadata	40
Gambar 4.7 Tampilan setelah menghapus metadata.	41
Gambar 4.8 Tampilan sistem pada perangkat <i>mobile</i>	42
Gambar 4.9 Ikon aplikasi pada daftar aplikasi di <i>mobile</i> (Android).	42
Gambar 4.10 Ikon aplikasi pada pencarian aplikasi di desktop (Windows 11). .	43
Gambar 4.11 Kode QR untuk membuka sistem secara langsung.....	43

DAFTAR TABEL

Tabel 2.1 Perbandingan penyimpanan <i>little endian</i> dan <i>big endian</i>	5
Tabel 2.2 <i>Marker</i> JPEG penting untuk metadata citra.	7
Tabel 2.3 Penanda (<i>marker</i>) metadata EXIF.	7
Tabel 2.4 Penanda format TIFF.	8
Tabel 2.5 Struktur data IFD.	8
Tabel 2.6 Tipe data <i>tag</i> pada IFD.	9
Tabel 2.7 Beberapa jenis <i>tag</i> pada TIFF dan EXIF.	10
Tabel 2.8 Beberapa pasangan <i>key-value</i> pada <i>manifest</i> PWA.	12
Tabel 2.9 Simbol pada diagram alir.	16
Tabel 2.10 Simpul pada <i>activity diagram</i>	17
Tabel 2.11 Simbol pada <i>use-case diagram</i>	19
Tabel 2.12 Rentang skor penilaian Lighthouse.	20
Tabel 2.13 Nilai bobot pada setiap audit untuk penghitungan aspek performa pada Lighthouse.	21
Tabel 2.14 Contoh pembagian bobot pada skala Likert.	22
Tabel 2.15 Rentang kriteria nilai <i>acceptance</i> berdasarkan skala Likert.	23
Tabel 3.1 Alur waktu penelitian.	25
Tabel 3.2 Daftar pernyataan pada <i>user acceptance testing</i>	33
Tabel 4.1 Beberapa pasangan <i>key-value web manifest</i> pada aplikasi.	39
Tabel 4.2 Pengujian fungsionalitas saat memilih citra.	44
Tabel 4.3 Pengujian fungsionalitas pada tabel daftar metadata citra.	45
Tabel 4.4 Pengujian fungsionalitas setelah menghapus metadata dari citra.	46
Tabel 4.5 Pengujian fungsionalitas aplikasi sebagai PWA.	46
Tabel 4.6 Skor pengujian Lighthouse aplikasi.	47
Tabel 4.7 Informasi <i>PWA Checklist</i> dari Lighthouse.	48
Tabel 4.8 Frekuensi pilihan jawaban setiap pernyataan.	49
Tabel 4.9 Penghitungan nilai T setiap pernyataan.	50
Tabel 4.10 Nilai <i>acceptance</i> setiap pernyataan.	50

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan dunia digital yang pesat sekarang membawa dampak yang beragam. Salah satunya adalah banyaknya layanan yang mendukung pengolahan citra digital (Apriliani, Hijjayanti, & Suhairoh, 2020) seperti situs media sosial. Karena ada banyak orang yang berinteraksi dengan media sosial, maka layanan tersebut seharusnya melindungi privasi penggunanya dengan mengamankan metadata citra digital yang diunggah oleh penggunanya. Meski tidak banyak, ada sekitar 31% layanan media sosial yang tidak melakukan penghapusan metadata citra (Gouert & Tsoutsos, 2022).

Citra digital adalah citra yang diambil menggunakan komputer. Citra digital dapat dilihat menggunakan komputer (Apriliani, Hijjayanti, & Suhairoh, 2020). Metadata citra digital adalah data tambahan yang tersembunyi dibalik citra digital. Metadata citra digital tidak terlihat saat citra dibuka untuk ditampilkan, melainkan membutuhkan aplikasi khusus. Metadata tersebut meliputi pengaturan kamera perangkat yang digunakan, modifikasi terakhir, dan bahkan data sensitif seperti data geolokasi (GPS) dan nomor serial yang berkaitan dengan perangkat atau kamera yang digunakan untuk mengambil citra digital tersebut. Banyak orang juga tidak menyadari keberadaan metadata citra ini karena alasan tersebut (Gouert & Tsoutsos, 2022).

Metadata citra tersebut biasanya ditaruh dalam sebuah format yang disebut EXIF (*Exchange Image File Format*), yaitu sebuah metode bagaimana memasukkan metadata pada citra dan audio digital. Ada beberapa cara membaca dan menghapus metadata citra digital seperti menggunakan *hex editor* secara manual (Wijayanto, Prabowo, & Harsadi, 2018) dan ExifTool namun alat itu biasanya digunakan untuk kebutuhan forensik dan kurang terjangkau bagi pengguna biasa (Gouert & Tsoutsos, 2022).

Progressive Web Application (kepanjangan dari PWA) adalah sebuah aplikasi web yang dapat memiliki fungsi lebih dari situs biasa. Pada platform *mobile*, PWA dapat diinstal dan terlihat seperti aplikasi terpisah. Meski berbasis

web, PWA tetap bisa diakses meski dalam keadaan luring (*offline*) (Nurwanto, 2019). PWA juga berjalan cepat meski dalam kondisi jaringan buruk dan dalam beberapa kasus dapat mengalahkan kecepatan aplikasi Android (Aminudin, Basren, & Nuryasin, 2019). Kemampuan jangkauan PWA juga terbilang cukup tinggi karena cukup dengan satu basis kode (*codebase*) untuk aplikasi web dan PWA. Beberapa perusahaan besar mengganti aplikasi mereka dengan aplikasi berbasis PWA dan mereka mendapatkan hasil yang sangat positif seperti peningkatan pengguna aktif dan penurunan ukuran aplikasi (Richard & LePage, 2020).

Berdasarkan uraian-uraian tersebut, maka akan dibuat aplikasi pembaca dan penghapus metadata EXIF pada citra digital berbasis *Progressive Web Apps*. Diharapkan bahwa aplikasi ini dapat berjalan pada seluruh perangkat, baik itu desktop maupun *mobile* dan mudah digunakan, serta dapat berjalan secara luring (*offline*), dan dapat dijangkau oleh semua orang.

1.2. Rumusan Masalah

Berdasarkan permasalahan dan latar belakang yang telah disebut, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi pembaca dan penghapus metadata citra digital dengan basis *Progressive Web Apps*?
2. Bagaimana efektivitas aplikasi pembaca dan penghapus metadata citra digital berbasis *Progressive Web Apps* berdasarkan masukan pengguna?

1.3. Batasan Masalah

Adapun batasan masalah penelitian ini adalah sebagai berikut:

1. Aplikasi ini memproses citra pengguna pada perangkat pengguna tanpa menggunakan server.
2. Aplikasi ini hanya memproses metadata EXIF. Artinya ada metadata yang tidak dapat diubah, seperti resolusi citra digital.
3. Aplikasi ini tidak mengubah data piksel citra yang ada termasuk konversi format penyimpanan citra digital.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah dan batasan masalah yang ada, maka tujuan penelitian ini adalah:

1. Merancang dan membangun aplikasi pembaca dan penghapus metadata EXIF pada citra digital berbasis *Progressive Web Apps*.
2. Mengetahui efektivitas aplikasi berdasarkan masukan dari pengguna.

BAB II

TINJAUAN PUSTAKA

2.1. Berkas Biner

Berkas biner adalah sebuah berkas yang terdiri atas banyak bit; unit terkecil pada data digital yang hanya memiliki nilai 0 atau 1 (The Linux Information Project, 2006). *Byte* adalah kumpulan 8 bit yang memiliki 256 permutasi untuk susunan 8 bit (tepatnya n bit memiliki 2^n permutasi) sehingga dapat merepresentasikan bilangan bulat 0 sampai 255. *Byte* adalah unit terkecil yang dapat dialamatkan oleh komputer modern. Kumpulan *byte* ini dapat digunakan untuk menaruh bilangan bulat dengan besaran yang bergantung pada banyaknya susunan *byte* yang digunakan (The Linux Information Project, 2007).

Untuk mempermudah pembacaan data biner oleh manusia, digunakan bilangan basis 16 atau heksadesimal. Heksadesimal dapat merepresentasikan 8 bit hanya dengan 2 angka. Angka heksadesimal dimulai dari 0 sampai 9 kemudian dilanjutkan dengan A, yang artinya 10, sampai dengan F, yang artinya 15. Notasi angka heksadesimal secara umum ditulis berdasarkan angkanya diikuti dengan basisnya (ditulis subskrip), misalnya 32_{16} (50_{10}), $2A_{16}$ (42_{10}), dan 20_{16} (32_{10}) (IONOS, 2021).

2.2. Jenis Data Biner

Ada beberapa cara yang dapat digunakan untuk membaca data biner. Cara-cara ini dikenal sebagai jenis data. Jenis data paling umum pada berkas biner adalah bilangan bulat, bilangan *floating-point*, dan susunan *byte* (*array*). Bilangan bulat memiliki 2 jenis, yaitu *unsigned* dan *signed*. Bilangan bulat *unsigned* artinya bilangan terendah yang dapat ditaruh adalah 0. Artinya kumpulan n -*byte* dapat merepresentasikan angka 0 sampai 2^{8n} . Bilangan bulat *signed* artinya bilangan terendah yang dapat ditaruh mulai dari angka -2^{8n-1} sampai dengan $2^{8n-1} - 1$. Bilangan *floating-point* adalah bilangan yang dapat menampung perkiraan dari bilangan riil dengan aturan khusus. Sistem penghitungan bilangan *floating-point* dan representasinya dalam kumpulan *byte* diatur oleh *IEEE Standard for Floating-*

Point Arithmetic (IEEE-754). Jenis bilangan *floating-point* yang paling umum digunakan adalah:

1. *Single-precision*, bilangan *floating-point* yang menggunakan 4 *byte* dan memiliki presisi sampai 24 digit biner, atau 7 digit desimal.
2. *Double-precision*, bilangan *floating-point* yang menggunakan 8 *byte* dan memiliki presisi sampai 53 digit biner, atau 15 digit desimal.

Kedua jenis *floating-point* tersebut sering disebut sebagai *binary32* dan *binary64* masing-masing (IEEE, 2019, hal. 23).

Endianness adalah aturan urutan membaca kumpulan *byte* pada posisi awalan tertentu. Ada 2 jenis *endianness* yang paling umum digunakan, yaitu *little endian* dan *big endian*. *Little endian* artinya nilai *byte* terkecil (*least-significant byte*) yang dibaca pertama. Sebaliknya, *big endian* artinya nilai *byte* terbesar (*most-significant byte*) yang dibaca pertama. Misalnya jika angka 12345678_{16} berada pada posisi awalan 0 pada berkas, maka urutan representasi setiap *byte* adalah sebagai berikut (Adiga, 2007).

Tabel 2.1 Perbandingan penyimpanan *little endian* dan *big endian*.

Posisi	<i>Little Endian</i>	<i>Big Endian</i>
0	78_{16}	12_{16}
1	56_{16}	34_{16}
2	34_{16}	56_{16}
3	12_{16}	78_{16}

2.3. Citra Digital

Citra digital adalah citra yang tersusun atas piksel dan memiliki panjang dan tinggi. Piksel adalah unit terkecil pada sebuah citra digital yang mengandung nilai intensitas atau nilai warna diskrit pada titik koordinat 2 dimensi tertentu (Gonzalez & Woods, 2018, hal. 18). Pengambilan citra digital dilakukan dengan menggunakan sensor yang sensitif terhadap cahaya dan memetakan posisinya pada titik koordinat tertentu. Proses tersebut disebut dengan *sampling*. Nilai intensitas piksel yang didapatkan pada proses *sampling* kemudian diubah ke bilangan bulat yang sesuai. Proses tersebut disebut *quantization* (Gonzalez & Woods, 2018, hal.

60-65). Setelah itu hasilnya yang disebut citra digital disimpan dalam bentuk berkas yang dapat dibaca oleh komputer.

2.4. Data Pribadi

Data pribadi adalah data tentang seseorang yang dapat digunakan untuk mengidentifikasi seorang individu, baik secara langsung maupun dengan kombinasi dari data lainnya. Ada 2 jenis data pribadi: yaitu data pribadi spesifik dan data pribadi umum. Data pribadi bersifat umum meliputi:

1. Nama lengkap
2. Jenis kelamin
3. Kewarganegaraan
4. Agama
5. Status perkawinan
6. Data pribadi lainnya yang dikombinasikan untuk mengidentifikasi seseorang (Indonesia, 2022).

2.5. Berkas Citra JPEG

JPEG, kepanjangan dari *Joint Photographic Experts Group*, adalah teknik kompresi citra digital yang mengorbankan sedikit kualitas untuk ukuran citra yang lebih kecil. JPEG dibuat pada tahun 1992 dan masih digunakan sampai sekarang. Kompresi JPEG umumnya memberi rasio kompresi sebanyak 10:1 dibandingkan dengan berkas citra tanpa kompresi. JPEG biasanya digunakan pada foto karena detail yang dihilangkan umumnya susah ditangkap oleh mata manusia (Wake, 2019).

Berdasarkan standar *CCITT Recommendation T.81*, berkas JPEG adalah berkas biner yang disusun oleh *marker*. *Marker* pada berkas JPEG berfungsi untuk mengidentifikasi bagian-bagian pada berkas JPEG. Semua *marker* diawali dengan *byte* FF₁₆ yang kemudian diikuti oleh *byte* kedua yang bukan 00₁₆ maupun FF₁₆. Setiap *marker* memiliki panjang berukuran 2-*byte* dengan urutan *big endian* yang diberikan setelah *marker*. Panjang *marker* sudah termasuk 2-*byte* panjangnya juga, sehingga panjang *marker* sebenarnya dikurangi 2. Ada beberapa *marker* khusus yang tidak memiliki panjang, seperti SOI dan EOI. Tabel 2.2 di bawah adalah beberapa jenis *marker* penting. Di sini, tanda bintang (*) pada simbol menunjukkan sebuah *marker* khusus tanpa panjang tertentu.

Tabel 2.2 Marker JPEG penting untuk metadata citra.

<i>Byte marker</i>	<i>Simbol marker</i>	Penjelasan
FF ₁₆ D8 ₁₆	SOI*	Awal berkas JPEG (<i>Start of Image</i>)
FF ₁₆ D9 ₁₆	EOI*	Akhir berkas JPEG (<i>End of Image</i>)
FF ₁₆ E _n ₁₆	APP _n	Data khusus aplikasi lainnya (<i>application segments</i>)
FF ₁₆ FE ₁₆	COM	Komentar tambahan pada citra.

2.6. Exchange Image File Format

EXIF, akronim dari *Exchange Image File Format*, adalah sebuah standar atau spesifikasi mengenai format tag untuk citra dan audio digital pada kamera digital yang dapat disisipkan sebagai metadata. Spesifikasi EXIF dibuat oleh *Japan Electronics and Information Technology Industries Association* (JEITA) pada tahun 1995. Tag-tag EXIF dapat berisi informasi mengenai konfigurasi kamera digital saat mengambil sebuah citra digital, model kamera digital, lokasi pengambilan citra digital tersebut, dan sebagainya.

Pada berkas citra JPEG, metadata EXIF dimasukkan pada *marker APP1* dan *APP2*. Metadata EXIF pada *marker APP1* ditandai dengan urutan-urutan *byte* berikut (Camera & Imaging Products Association, 2019):

Tabel 2.3 Penanda (*marker*) metadata EXIF.

Posisi	<i>Byte</i>	Deskripsi
n	FF ₁₆ E1 ₁₆	<i>Marker APP1.</i>
$n + 2$	XX ₁₆ YY ₁₆	Panjang <i>marker</i> sebanyak 2 <i>bytes</i> dengan urutan <i>big endian</i> .
$n + 4$	45 ₁₆ 78 ₁₆ 69 ₁₆ 66 ₁₆ 00 ₁₆ 00 ₁₆	Tulisan “Exif” diikuti dengan 2 nilai 0 atau <i>null byte</i> . Digunakan sebagai penanda bahwa <i>marker APP1</i> tersebut diisi dengan metadata EXIF.

Setelah penanda (*header*) tersebut, struktur data metadata EXIF akan mengikuti format TIFF.

2.7. Tag Image File Format

TIFF, akronim dari *Tag Image File Format* adalah sebuah format berkas yang digunakan untuk menyimpan citra beserta metadatanya. EXIF sendiri menggunakan format TIFF revisi 6 untuk menaruh metadata citra, tanpa data citra itu sendiri. Setelah penanda metadata EXIF langsung diikuti oleh penanda TIFF. TIFF sendiri terdiri atas penanda *endianness*, sebuah nilai konstan, dan posisi dimana informasi mengenai citra disimpan (Adobe Inc., 1992).

Tabel 2.4 Penanda format TIFF (Adobe Inc., 1992).

Posisi	Byte	Deskripsi
$n + 10$	$49_{16} 49_{16}$ atau $4D_{16} 4D_{16}$	Penanda <i>endianness</i> . Jika awalnya adalah “II” ($49_{16} 49_{16}$) maka urutan pembacaannya adalah <i>little endian</i> . Jika awalnya adalah “MM” ($4D 4D$) maka urutan pembacaannya adalah <i>big endian</i> .
$n + 12$	$002A_{16}$	Nilai 42, dibaca sebagai bilangan bua 2-byte dengan urutan <i>endianness</i> yang ditentukan sebelumnya.
$n + 14$	4-byte	Posisi <i>Image File Directory</i> (IFD) awal, atau IFD 0.

Setiap IFD memiliki struktur sebagai berikut:

Tabel 2.5 Struktur data IFD (Adobe Inc., 1992).

Posisi	Byte	Deskripsi
n	2-byte	Banyak entri pada IFD ini (m).
Untuk setiap entri pada IFD, i dimulai dari 0 sampai dengan $m - 1$.		
$n + 2 + 12i$	2-byte	Jenis <i>tag</i> yang mengidentifikasi entri ini.
$n + 2 + 12i + 2$	2-byte	Jenis nilai <i>tag</i> /tipe data.
$n + 2 + 12i + 4$	4-byte	Banyak nilai pada <i>tag</i> ini.
$n + 2 + 12i + 8$	4-byte	Posisi di mana nilai <i>tag</i> ini diletakkan. Jika nilainya dapat muat dalam 4-byte maka bagian ini berisi nilai <i>tag</i> , bukan posisinya.
Selesai untuk setiap entri pada IFD		
$n + 2 + 12m$	4-byte	Posisi IFD berikutnya.

Setiap data pada IFD memiliki jenis nilai yang mungkin sebagai berikut:

Tabel 2.6 Tipe data *tag* pada IFD (Adobe Inc., 1992).

Nomor Jenis <i>Tag</i>	Nama <i>Tag</i>	Deskripsi
1	BYTE	Bilangan bulat 1- <i>byte</i> .
2	ASCII	Daftar <i>bytes</i> dengan panjang tertentu yang diperlakukan seperti teks yang dapat dibaca oleh manusia. <i>Byte</i> terakhir harus bernilai 0.
3	SHORT	Bilangan bulat 2- <i>byte</i> .
4	LONG	Bilangan bulat 4- <i>byte</i> .
5	RATIONAL	2 bilangan bulat 4- <i>byte</i> yang merepresentasikan pecahan. Bilangan bulat pertama adalah pembilang dan bilangan bulat kedua adalah penyebut.
6	SBYTE	Bilangan bulat 1- <i>byte</i> yang dapat menampung bilangan negatif.
7	UNDEFINED	Daftar <i>bytes</i> dengan panjang tertentu.
8	SSHORT	Bilangan bulat 2- <i>byte</i> yang dapat menampung bilangan negatif.
9	SLONG	Bilangan bulat 4- <i>byte</i> yang dapat menampung bilangan negatif.
10	SRATIONAL	2 bilangan bulat 4- <i>byte</i> yang merepresentasikan pecahan. Bilangan bulat pertama adalah pembilang dan bilangan bulat kedua adalah penyebut. Dapat menampung pecahan negatif.
11	FLOAT	Bilangan riil dalam bentuk <i>floating-point</i> 4- <i>byte</i> (<i>single-precision</i>).
12	DOUBLE	Bilangan riil dalam bentuk <i>floating-point</i> 8- <i>byte</i> (<i>double-precision</i>).

Ada banyak jenis *tag* IFD, baik yang didefinisikan oleh TIFF 6.0 sendiri maupun dari EXIF.

Tabel 2.7 Beberapa jenis *tag* pada TIFF dan EXIF (Adobe Inc., 1992; Camera & Imaging Products Association, 2019).

Jenis <i>Tag</i>	Nama <i>Tag</i>	Jenis Data	Deskripsi
270	ImageDescription	ASCII	Deskripsi citra, jika ada.
271	Make	ASCII	Merek dari perangkat.
272	Model	ASCII	Model perangkat.
305	Software	ASCII	Perangkat lunak yang digunakan untuk membuat citra.
33434	ExposureTime	RATIONAL	Berapa lama sensor mengambil cahaya, dalam detik.
38687	DateTimeOriginal	ASCII	Kapan citra diambil.
36868	DateTimeDigitized	ASCII	Kapan citra diubah jadi format digital.
37382	SubjectDistance	RATIONAL	Jarak subjek terhadap kamera.

2.8. Comma-Separated Values

Comma-Separated Values, kepanjangan dari CSV, adalah sebuah format berkas teks sederhana yang berfungsi mendefinisikan data dalam bentuk tabel. CSV biasanya digunakan untuk berbagi data tabel dengan aplikasi-aplikasi *spreadsheet*. CSV tidak memiliki spesifikasi yang formal, sehingga beberapa aplikasi menulis dan membacanya sedikit berbeda. Format CSV cukup sederhana, yaitu (Shafranovich, 2005):

1. Baris awal adalah *header* tabel.
2. Sesuai dengan namanya, tanda koma menandakan pemisah antar kolom.
3. Baris baru (teks) berarti baris baru pada tabel.
4. Baris baru (teks) menggunakan *byte* 0D₁₆ 0A₁₆ atau sering disebut CRLF (singkatan dari *Carriage Return, Line Feed*).

5. Setiap kolom dapat diapit tanda kutip dua, namun kolom yang berisi tanda koma harus diapit tanda kutip dua.

2.9. HTML, CSS, dan JavaScript

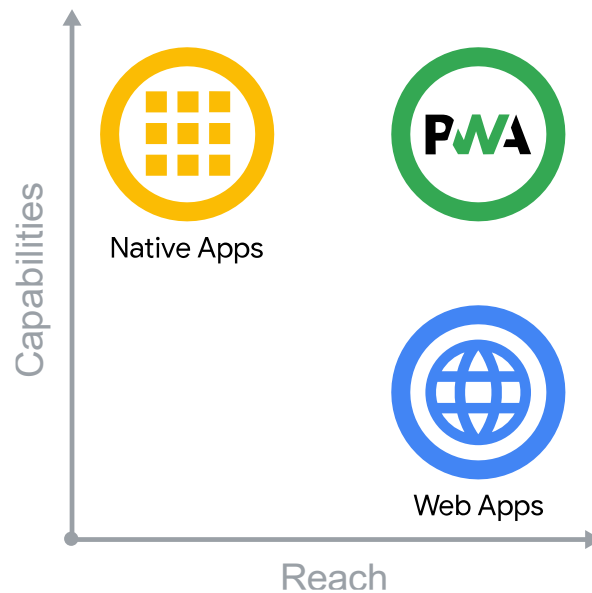
Hypertext Markup Language (HTML) adalah sebuah bahasa markup untuk mendeskripsikan dokumen web (Web Hypertext Application Technology Working Group, 2023). Awalnya HTML dibuat oleh Tim Berners-Lee pada tahun 1989 yang kemudian dikembangkan oleh *World Wide Web Consortium* (W3C) pada 1995 dan kemudian pada 2004 dibentuklah *Web Hypertext Application Technology Working Group* (WHATWG) untuk bertanggung jawab atas perkembangan bahasa HTML (Saputra & Faz, 2019, hal. 2).

Cascading Style Sheet (CSS) adalah sebuah bahasa yang mendeskripsikan bagaimana sebuah halaman web di tampilkan. CSS menangani bagaimana warna, tata letak, dan bentuk huruf yang harus ditampilkan oleh aplikasi penjelajah web (World Wide Web Consortium, 2016).

JavaScript adalah bahasa pemrograman yang digunakan untuk mendefinisikan perilaku halaman web saat pengguna berinteraksi dengan halaman web tersebut. JavaScript berjalan pada aplikasi *browser* pengguna dan dapat dimasukkan langsung dalam HTML (Siahaan & Sianipar, 2018, hal. 1-3).

2.10. Progressive Web Application

Progressive Web Application (kepanjangan dari PWA) adalah sebuah aplikasi web yang memiliki fungsi lebih dari sekedar situs web biasa. PWA dapat diinstal dan tidak membutuhkan koneksi internet. PWA juga dapat berjalan baik pada platform desktop dan *mobile* serta memiliki keamanan yang canggih. Artinya, PWA dapat menjadi aplikasi yang seolah-olah terintegrasi pada platform tersebut tetapi saat yang bersamaan juga merupakan aplikasi web. PWA juga memiliki statistik penggunaan dan jangkauan yang lebih baik (Richard & LePage, 2020). *Progressive Web Apps* dibuat menggunakan teknologi web yang sudah ada seperti HTML, CSS, dan JavaScript.



Gambar 2.1 Perbandingan kemampuan dan jangkauan jenis-jenis aplikasi (Richard & LePage, 2020).

Pada PWA, sebuah berkas disebut *manifest* digunakan untuk mendefinisikan nama aplikasi, ikon aplikasi, pintasan-pintasan yang tersedia, dan sebagainya. Berkas *manifest* ini didefinisikan dengan sintaks JSON (*JavaScript Object Notation*) dengan *key-value pairs* sebagai berikut (LePage, Beaufort, & Steiner, 2022).

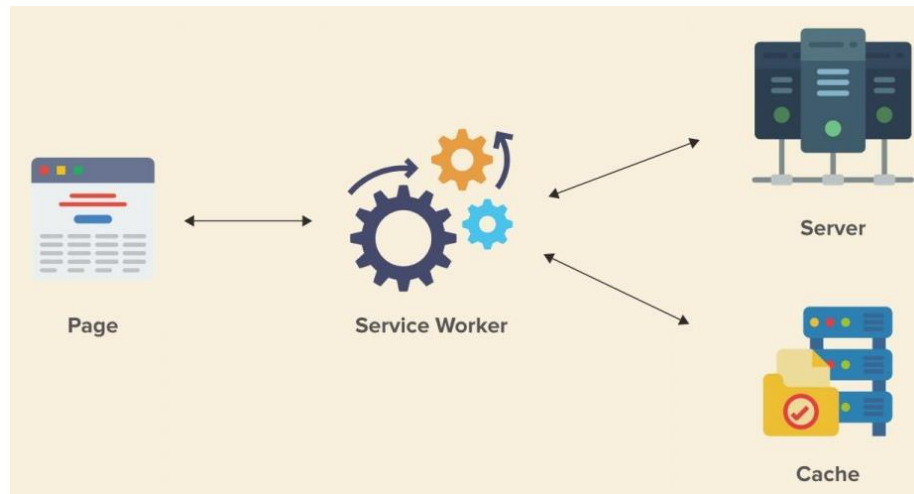
Tabel 2.8 Beberapa pasangan *key-value* pada *manifest* PWA

Key	Deskripsi Isi Nilai
short_name	Nama singkat aplikasi.
name	Nama lengkap aplikasi.
icons	Daftar yang berisi ikon yang digunakan berdasarkan jenis dan ukuran dimensi citra.
description	Deskripsi aplikasi.
start_url	Lokasi (relatif pada aplikasi web) yang akan dibuka ketika aplikasi dijalankan.
display	Bagaimana aplikasi di tampilkan.
shortcuts	Pintasan-pintasan yang tersedia pada aplikasi.

2.11. Service Worker

Service Worker adalah sebuah kode JavaScript yang berjalan pada latar belakang. *Service Worker* dapat digunakan untuk mencegat permintaan berkas pada

jaringan, mengirimkan notifikasi, dan melakukan sinkronisasi pada latar belakang. Contohnya, pada instalasi PWA, *Service Worker* bertugas untuk melakukan *caching* berkas-berkas yang dibutuhkan untuk menjalankan aplikasi secara *offline*. Kemudian, *Service Worker* dapat mencegat permintaan browser dan melanjutkan permintaan berkas dari jaringan jika pengguna sedang *online*, atau mengembalikan data dari *cache* jika pengguna sedang *offline* (Vu, 2020).



Gambar 2.2 Posisi *Service Worker* pada aplikasi web

2.12. Metodologi Agile

Metodologi *agile* merupakan metodologi pengembangan yang bersifat kontinu atau iteratif. Berbeda dengan metodologi tradisional seperti *waterfall*, metodologi *agile* dapat beradaptasi cepat terhadap perubahan kebutuhan, sedangkan perubahan kebutuhan pada *waterfall* artinya membuat ulang sistem dari awal. Metodologi *agile* mulai populer karena sistem dapat dibuat dengan cepat dan fitur dapat ditambahkan atau diubah saat dalam tahap pengembangan atau setelah sistem di buat (O'Regan, Concise Guide to Software Testing, 2019).



Gambar 2.3 Siklus metodologi *agile* (Eka, 2022).

Secara garis besar, metodologi *agile* dibagi menjadi beberapa bagian, yaitu (O'Regan, Concise Guide to Software Engineering: From Fundamentals to Application Methods, 2017; Eby, 2016):

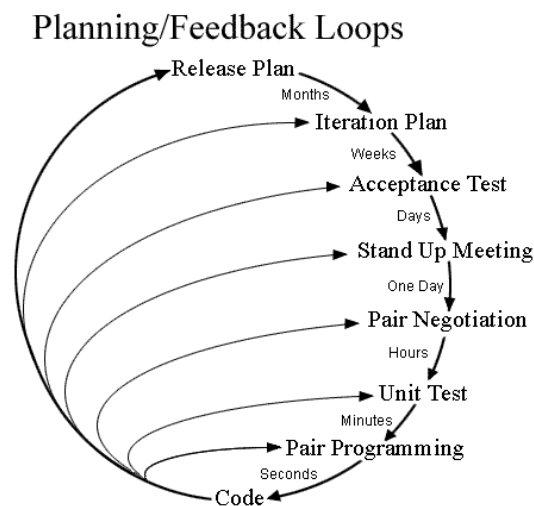
1. Kebutuhan (*requirement*) dan rancangan (*design*) dievaluasi setiap iterasi agar sesuai dengan kebutuhan konsumen berdasarkan masukan (*review/feedback*).
2. Kode pengujian (*testing*) dituliskan terlebih dahulu sebelum implementasi (*development*). Ini disebut *test-driven development*.
3. Sebelum aplikasi dipublikasi (*deploy*), aplikasi harus lolos pengujian-pengujian (otomatis) yang ada. Semua perubahan baik kecil maupun besar harus diintegrasikan terus menerus (*continuous integration*).
4. Setelah aplikasi dipublikasi, maka akan dilihat lagi masukan dari pengguna untuk mengevaluasi kebutuhan dan rancangan.

2.13. Extreme Programming

Extreme Programming (ditulis *eXtreme Programming*, disingkat XP) merupakan salah satu metodologi *agile* yang dikembangkan oleh Kent Beck pada 1999. Metode pengembangan ini memiliki siklus pengembangan yang singkat, mengutamakan kesederhanaan, berfokus ke pembuatan perangkat lunak, dan dapat beradaptasi dengan cepat terhadap perubahan berdasarkan masukan dari konsumen (*feedback*). Metodologi ini juga mementingkan pengujian berkala dan integrasi kontinu, sehingga masalah pada perangkat lunak dapat ditangkap dan diatasi lebih awal dan perangkat lunak dapat dirilis dalam waktu cepat.

Pada metodologi XP, beberapa bagian pada tahapan *agile* ditingkatkan pada tingkat ekstrem, seperti (Beck & Andres, 2004):

1. Satu siklus rilis perangkat lunak paling lambat harus selesai dalam waktu beberapa bulan. Ini bertujuan agar masukan dari konsumen terus berjalan dan bersifat kontinu.
2. Pada perancangan sistem, rancang sistem sederhana terlebih dahulu berdasarkan kebutuhan secara umum. Kemudian, secara perlahan mulai merancang hal-hal yang lebih spesifik.
3. Pengujian fitur dituliskan terlebih dahulu dulu sebelum mengimplementasikan fitur tersebut. Ini bertujuan untuk mengetahui masalah pada perangkat lunak secara dini dan membatasi batasan masalah pada fitur.
4. Semua perubahan, baik itu besar maupun kecil, setelah lolos pada tahap pengujian, harus diintegrasikan pada perangkat lunak, meski itu dapat terjadi dalam waktu beberapa jam sehari.



Gambar 2.4 Tahapan perencanaan sampai masukan pada metodologi XP (Wells, 2013)




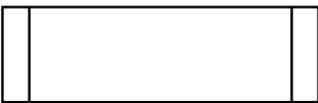
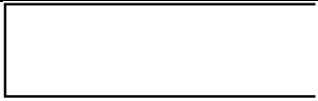

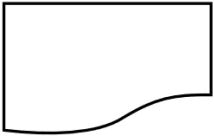
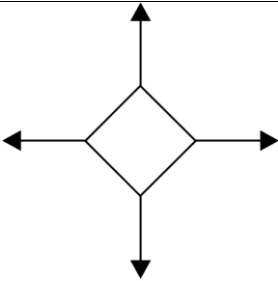
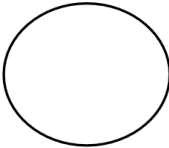
2.14. Flowchart

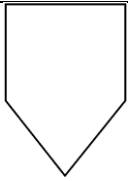
Diagram alir (*flowchart*) adalah sebuah diagram yang menggambarkan cara kerja sebuah (bagian dari) program. Diagram alir mempermudah membaca cara kerja program dibandingkan dengan membaca cara kerja program secara tekstual. Pada Diagram alir, kotak dengan berbagai bentuk digunakan untuk menggambarkan jenis-jenis operasi yang berbeda. Garis yang disebut garis alir

(*flow lines*) digunakan untuk menyambung kotak-kotak tersebut. Diagram alir biasanya dibaca dari atas ke bawah dan dari kiri ke kanan.

Berikut adalah simbol-simbol pada diagram alir beserta penjelasannya (Chaudhuri, 2020):

Tabel 2.9 Simbol pada diagram alir.

Simbol	Nama	Deskripsi
	<i>Terminal</i>	Mendefinisikan awalan dan akhiran sebuah program.
	<i>Input/Output</i>	Digunakan untuk menjelaskan operasi masukan atau keluaran.
	<i>Computer Processing</i>	Menjelaskan proses yang dilakukan program.
	<i>Predefined Processing</i>	Menjelaskan proses yang sudah dijelaskan pada diagram alir yang sudah ada.
	<i>Comment</i>	Digunakan untuk menuliskan penjelasan tambahan.
	<i>Flow line</i>	Digunakan untuk menghubungkan simbol.
	<i>Document Input/Output</i>	Menjelaskan bahwa masukan berasal dari dokumen atau keluaran ditulis ke dokumen.
	<i>Decision</i>	Digunakan saat proses memerlukan keputusan sebelum melanjutkan ke proses lainnya.
	<i>On-page Connector</i>	Menghubungkan diagram alir pada halaman yang sama.

Simbol	Nama	Deskripsi
	<i>Off-page Connector</i>	Menghubungkan diagram alir pada halaman yang berbeda.

2.15. Unified Modeling Language



Unified Modeling Language, kepanjangan dari UML, adalah sebuah spesifikasi bagaimana memodelkan sebuah sistem perangkat lunak. UML memudahkan komunikasi antara perancang sistem dan pengembang perangkat lunak mengenai sistem perangkat lunak yang di buat. UML menggunakan diagram sebagai bahasa pemodelan sebuah sistem. Ada banyak jenis diagram yang didefinisikan oleh UML, beberapa di antaranya yaitu *Activity Diagram* dan *Use-case Diagram*.



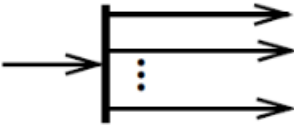
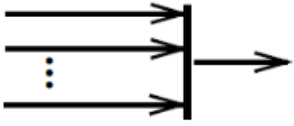
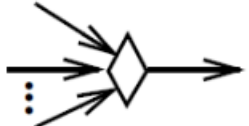
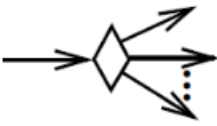
Activity Diagram adalah sebuah diagram yang mendefinisikan perilaku sistem secara berurutan. *Activity Diagram* harus setidaknya memiliki simpul-simpul berikut:

1. Satu atau lebih simpul aksi.
2. Satu atau lebih simpul kontrol yang berfungsi sebagai titik awal *activity*.
3. Satu atau lebih simpul kontrol yang berfungsi sebagai titik akhir seluruh *activity*.
4. Nol atau lebih simpul kontrol lainnya.

Berikut adalah simbol-simbol simpul pada *activity diagram*:

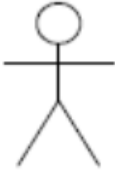



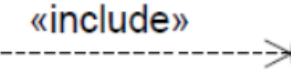
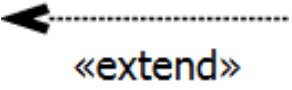
Tabel 2.10 Simpul pada *activity diagram*.

Simbol	Nama	Deskripsi
	Aksi (<i>action</i>)	Mendeskripsikan hal yang dilakukan.
	Simpul awal <i>activity</i> (<i>initial node</i>)	Titik awal mulai sebuah <i>activity</i> .

Simbol	Nama	Deskripsi
	Simpul akhir <i>activity</i> (<i>final node</i>)	Titik akhir dari sebuah aliran eksekusi dan menghentikan seluruh <i>activity</i> .
	Simpul akhir alir (<i>flow final node</i>)	Titik akhir dari sebuah aliran eksekusi tetapi <i>activity</i> tetap berlanjut.
	Simpul mulai percabangan (<i>fork node</i>)	Memisah aliran menjadi 2 atau lebih aliran yang berjalan secara bersamaan, masing-masing menjalankan aksi yang berbeda.
	Simpul gabung percabangan (<i>join node</i>)	Menunggu aliran-aliran aksi yang berjalan mencapai simpul ini kemudian berlanjut ke aksi selanjutnya.
	Simpul gabung (<i>merge node</i>)	Menggabungkan aliran menjadi 1 aliran yang akan menjalankan aksi berikutnya. Berbeda dengan <i>join node</i> , <i>merge node</i> tidak menunggu seluruh aliran selesai.
	Simpul pilihan (<i>decision node</i>)	Memilih salah 1 dari 2 atau lebih aliran untuk dijalankan berdasarkan kondisi yang tertulis.

Use-case Diagram adalah sebuah diagram yang digunakan untuk menggambarkan apa yang sistem seharusnya lakukan. *Use-case diagram*, menggambarkan interaksi aktor terhadap *use-case* yang ada pada sebuah sistem. Berikut adalah simbol-simbol pada *use-case diagram* (Object Management Group, 2017):

Tabel 2.11 Simbol pada *use-case diagram*.

Simbol	Nama	Deskripsi
 Customer	Aktor	Notasi aktor menggunakan <i>stick-man</i> . Aktor akan berinteraksi pada sistem dalam bentuk <i>use-case</i> .
	Penghubung	Penghubung antara aktor dan <i>use-case</i> yang aktor dapat lakukan.
 Withdraw	<i>Use-case</i>	Notasi <i>use-case</i> , apa yang sistem bisa lakukan.
	Kepemilikan	Notasi kepemilikan <i>use-case</i> yang berbentuk kotak (pada contoh gambar, <i>use-case</i> dimiliki oleh <i>DepartmentStore</i>) digunakan untuk menjelaskan bahwa sebuah <i>use-case</i> merupakan bagian dari sebuah sistem tertentu.
	<i>Include</i>	Notasi <i>include</i> digunakan untuk mendefinisikan bahwa sebuah <i>use-case</i> membutuhkan <i>use-case</i> yang lain agar dapat didefinisikan sebelumnya.
	<i>Extend</i>	Notasi <i>extend</i> , yaitu menambahkan <i>use-case</i> tambahan dari <i>use-case</i> yang sudah ada.

2.16. Pengujian *Black Box*

Pengujian *black box* (*Black Box Testing*) merupakan pengujian yang dilakukan untuk memastikan sistem berfungsi sesuai dengan spesifikasi. Pengujian *black box* dilakukan tanpa mengetahui cara kerja sebuah sistem. Ada banyak jenis pengujian *black box*, beberapa diantaranya yaitu:

1. *Functionality testing*, yaitu menguji fungsionalitas sistem apakah sudah bekerja seperti semestinya sesuai kebutuhan.
2. *Equivalence partitioning*, yaitu membagi nilai pengujian menjadi kelas-kelas dengan rentang yang sama besar kemudian mengambil nilai tengah dari kelas tersebut.
3. *Boundary value Analysis*, mirip seperti *equivalence partitioning* tetapi memilih nilai dari ujung-ujung batas kelas, bukan nilai tengah kelas.
4. *Use-case testing*, yaitu pengujian bagaimana aktor akan berinteraksi terhadap sistem berdasarkan *use-case diagram* (O'Regan, Concise Guide to Software Testing, 2019).

2.17. Pengujian Lighthouse

Lighthouse adalah alat yang dikembangkan oleh Google untuk menguji sebuah aplikasi web, termasuk PWA. Lighthouse menguji aspek performa, praktik-praktik terbaik (*best practices*), aksesibilitas, optimisasi mesin pencari (*Search Engine Optimization*, SEO), dan bagaimana PWA pada situs web diimplementasikan, kemudian memberi skor 0-100 pada tiap-tiap aspek tersebut (Rochim, Rahmatulloh, Akbar, & Rizal, 2023).

Tabel 2.12 Rentang skor penilaian Lighthouse

Rentang Skor	Arti Skor	Warna
0-49	Lambat/kurang baik	Merah
50-89	Rata-rata	Kuning
90-100	Cepat/sangat baik	Hijau

Penghitungan skor Lighthouse cukup kompleks dan bergantung dari aspek-aspek yang akan diuji:

1. Performa

Penghitungan skor performa pada Lighthouse dilakukan dengan mengonversi nilai setiap audit menjadi *metric score* yang kemudian dirata-ratakan dengan bobot (*weighted average*) pada setiap audit.

Tabel 2.13 Nilai bobot pada setiap audit untuk penghitungan aspek performa pada Lighthouse.

Audit	Satuan	Bobot	Deskripsi
<i>First Contentful Paint</i>	Mili detik	10%	Seberapa lama waktu yang dibutuhkan untuk memulai menampilkan halaman web.
<i>Speed Index</i>	Mili detik	10%	Seberapa cepat halaman ditampilkan saat sedang dimuat.
<i>Largest Contentful Paint</i>	Mili detik	25%	Berapa lama waktu yang dibutuhkan untuk menampilkan konten terbesar.
<i>Total Blocking Time</i>	Mili detik	30%	Waktu kumulatif mulai dari FCP dimana masukan pengguna tidak bisa diproses.
<i>Cumulative Layout Shift</i>	Tanpa satuan	25%	Seberapa “banyak” elemen bergeser pada halaman secara keseluruhan.

Konversi nilai audit ke *metric score* dilakukan dengan menghitung distribusi kumulatif *log-normal* menggunakan model dari *HTTP Archive* dan menentukan posisi nilai pada kurva distribusi kumulatif tersebut (Wagner & Kenny, 2023).

2. Aksesibilitas

Skor keseluruhan aspek aksesibilitas dihitung berdasarkan rata-rata berbobot (*weighted average*) dari semua audit aksesibilitas yang diuji. Bobot-bobot setiap audit didasarkan pada *aXe user impact assessment*. Setiap audit hanya bisa “lulus” atau “tidak lulus”, sehingga jika hanya ada beberapa elemen yang “lulus” pada audit tertentu, maka audit tersebut dianggap “tidak lulus” (Kenny & Emelianova, 2023).

3. Praktik Terbaik (*Best Practice*)

Skor pada aspek praktik terbaik dihitung berdasarkan rata-rata dari semua audit yang diuji.

4. Optimisasi Mesin Pencari (SEO)

Skor pada aspek optimisasi mesin pencari dihitung berdasarkan rata-rata dari semua audit yang diuji. Skor SEO yang tinggi akan muncul lebih teratas pada mesin pencari (van der Schee, 2019).

5. PWA

Tidak ada skor pada aspek ini, tetapi hasil aspek ini dibagi menjadi 3 grup yang berbeda dengan auditnya masing-masing: Cepat dan tangkas (*fast and reliable*), dapat diinstal (*installable*), dan dioptimisasi untuk PWA (*PWA optimized*).

2.18. User Acceptance Testing

User Acceptance Testing, kepanjangan dari UAT, adalah salah satu jenis pengujian sistem. UAT dilakukan oleh pengguna agar memastikan bahwa sistem memenuhi ekspektasi dan kebutuhan pengguna (O'Regan, Concise Guide to Software Testing, 2019). Pada kuesioner *acceptance testing*, salah satu skala yang digunakan adalah Skala Likert.

Skala Likert adalah skala yang biasanya digunakan untuk mengukur pendapat seseorang terhadap pertanyaan yang diajukan. Pada pertanyaan positif, skala yang diberikan adalah 1 (untuk respon “salah”) sampai 5 (untuk respon ”benar”). Sedangkan pada pertanyaan negatif, skala tersebut di balik (Pranatawijaya, Widiatry, Priskila, & Putra, 2019). Contohnya seperti berikut:

Tabel 2.14 Contoh pembagian bobot pada skala Likert (Setiono & Riwinoto, 2015)

Tanggapan	Bobot
Sangat Tidak Setuju	1
Tidak Setuju	2
Netral	3
Setuju	4
Sangat Setuju	5

Untuk menghitung nilai *acceptance* untuk setiap jawaban, total nilai setiap jawaban harus dihitung terlebih dahulu dengan rumus berikut:

$$T = \sum_{i=1}^5 t_i p_i$$

Keterangan:

1. T – Total nilai setiap jawaban.
2. t_i – Banyak responden yang memilih tanggapan ke- i .
3. p_i – Nilai bobot tanggapan ke- i .

Kemudian, nilai *acceptance* setiap jawaban dapat dihitung dengan rumus berikut:

$$A\% = \frac{T}{nh} \times 100$$

Keterangan:

1. $A\%$ – Nilai *acceptance* setiap jawaban dalam persen.
2. T – Total nilai setiap jawaban.
3. n – Banyak responden.
4. h – Bobot nilai tertinggi setiap jawaban (Setiono & Riwinoto, 2015)

Tabel 2.15 Rentang kriteria nilai *acceptance* berdasarkan skala Likert

Kriteria	Rentang Nilai
Buruk	$0\% \leq N \leq 20\%$
Kurang Baik	$20\% < N \leq 40\%$
Cukup	$40\% < N \leq 60\%$
Baik	$60\% < N \leq 80\%$
Sangat Baik	$80\% < N \leq 100\%$

2.19. Penelitian Terkait

Pada penelitian yang berjudul “Optimalisasi Penyusutan EXIF Metadata Dengan Teknik Substitusi NULL Value Pada Kasus Keamanan Citra Digital”, dilakukan ekstraksi metadata EXIF ke bentuk teks yang mudah dibaca kemudian melakukan substitusi *null-value* ke metadata tersebut. Karena mereka melakukannya secara manual menggunakan *hex editor* dan bantuan aplikasi dari

penelitian sebelumnya, mereka berharap bahwa ada aplikasi yang bisa melakukan hal ini secara otomatis dan mudah digunakan di masa mendatang (Wijayanto, Prabowo, & Harsadi, 2018).

Pada penelitian yang berjudul “Analisis Keaslian Citra Dengan Menggunakan EXIF Metadata”, dilakukan perbandingan antara 2 citra untuk tujuan forensik. Citra yang sudah dimanipulasi memiliki metadata yang berbeda dibandingkan dengan citra yang belum dimanipulasi. *Python* digunakan sebagai bahasa pemrograman untuk melaksanakan penelitian tersebut. Metodologi penelitiannya menggunakan metodologi *Extreme Programming*. Hasilnya adalah, citra yang sudah dimanipulasi cenderung memiliki metadata yang lebih banyak, termasuk aplikasi yang digunakan untuk mengubah citra aslinya (Apriliani, Hijjayanti, & Suhairoh, 2020).

Pada penelitian yang berjudul “Dirty Metadata: Understanding A Threat to Online Privacy”, dituliskan bahwa metadata citra berbahaya jika tidak dihapus. Salah satu bahaya yang disebutkan adalah metadata tersebut dapat digunakan oleh oknum jahat yang dapat mengancam privasi pengguna. Penelitian tersebut meneliti berbagai situs bagaimana metadata citra ditangani. Meski mayoritas situs yang mereka uji melakukan penghapusan metadata, tetapi tidak sedikit juga yang membiarkan metadata tersebut dalam citra (Gouert & Tsoutsos, 2022).

Penelitian yang berkaitan mengenai *progressive web apps* ada 2. Pertama yaitu penelitian yang berjudul “Penerapan Progressive Web Application (PWA) pada E-Commerce”. Penelitian tersebut membuat aplikasi jual beli laptop berbasis PWA sehingga aplikasi tetap terbuka meski dalam keadaan luring (Nurwanto, 2019). Penelitian kedua yaitu penelitian yang berjudul “Perancangan Sistem Repositori Tugas Akhir Menggunakan Progressive Web App (PWA)”. Penelitian tersebut membuat aplikasi repositori tugas akhir menggunakan basis PWA. Berdasarkan pengujian Lighthouse untuk PWA, aplikasi tersebut mendapatkan skor 92.4 dari skor maksimal 99, artinya aplikasi mereka tergolong sangat baik (Aminudin, Basren, & Nuryasin, 2019).