

**ANALISIS PERBANDINGAN DUA DAN EMPAT LEVEL  
GRAYSCALE SERTA ARUCO MARKER DALAM  
PENGEMBANGAN CDP PADA SQR**

**SKRIPSI**



**Disusun oleh:**

**Ade Firmansyah  
19/440303/TK/48630**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA  
YOGYAKARTA  
2023**

## **HALAMAN PENGESAHAN**

### **ANALISIS PERBANDINGAN DUA DAN EMPAT LEVEL GRAYSCALE SERTA ARUCO MARKER DALAM PENGEMBANGAN CDP PADA SQR**

#### **SKRIPSI**

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana Teknik  
pada Departemen Teknik Elektro dan Teknologi Informasi  
Fakultas Teknik  
Universitas Gadjah Mada

Disusun oleh:

**Ade Firmansyah  
19/440303/TK/48630**

Telah disetujui dan disahkan

Pada tanggal .....

Dosen Pembimbing I

Dosen Pembimbing II

**Syukron Abu Ishaq Alfarizi, S.T., Ph.D.**  
**1111 9920 5202 10 1 102**

**«Nama Dosen»  
«NIP xxxxxxx»**

## **PERNYATAAN BEBAS PLAGIASI**

Saya yang bertanda tangan di bawah ini :

Nama :  
NIM :  
Tahun terdaftar :  
Program Studi :  
Fakultas : Teknik Universitas Gadjah Mada

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, tanggal-bulan-tahun

Materai Rp10.000

(Tanda tangan)

Nama Mahasiswa  
NIM

## **HALAMAN PERSEMBAHAN**

Tugas akhir ini kupersembahkan kepada kedua orang tuaku. Kupersembahkan pula kepada keluarga dan teman-teman semua, serta untuk bangsa, negara, dan agamaku.

[contoh]

## **KATA PENGANTAR**

### **[SAMPLE]**

Puji syukur ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik. Dalam hal penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. <isi dengan nama Kadep>
2. <isi dengan nama Sekdep>
3. <isi dengan nama Dosen Pembimbing>
4. Kedua Orang Tua, kakak, dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
5. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua, aamiin. [Contoh]

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
PERNYATAAN BEBAS PLAGIASI .....	iii
HALAMAN PERSEMBAHAN .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vi
DAFTAR TABEL .....	ix
DAFTAR GAMBAR .....	x
DAFTAR SINGKATAN.....	xi
INTISARI.....	xii
ABSTRACT .....	xiii
BAB I Pendahuluan .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Penelitian .....	2
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan.....	4
BAB II Tinjauan Pustaka dan Dasar Teori .....	5
2.1 Tinjauan Pustaka .....	5
2.1.1 Apakah Pola Deteksi Duplikat (CDP) dapat Disalin .....	5
2.1.1.1 Prinsip Degradasi Informasi.....	5
2.1.1.2 Definisi Teoritis dari Sistem Autentikasi CDP .....	6
2.1.1.3 Komponen dari Detektor .....	7
2.1.2 Detekti Pembajakan menggunakan SQR .....	7
2.1.2.1 Struktur dari SQR.....	8
2.1.2.2 Pembuatan dan Pencetakan SQR .....	9
2.1.2.3 Autentikasi SQR .....	9
2.1.3 Autentikasi Digital menggunakan CDP.....	10
2.1.4 Pencetakan Variasi CDP .....	10
2.1.5 Autentikasi CDP menggunakan Perangkat Seluler.....	10
2.2 Dasar Teori .....	10
2.2.1 Kode QR.....	10
2.2.1.1 Bagaimana Kode QR Bekerja.....	11
2.2.1.2 Versi Kode QR .....	11
2.2.1.3 Koreksi Kesalahan Kode QR .....	12
2.2.2 Deteksi Pola Duplikat (CDP).....	12

2.2.3	Lokalisasi Objek dengan Pengenalan Pola.....	13
2.2.4	ArUco <i>Marker</i> .....	13
2.2.5	Koefisien Jarak .....	14
2.2.5.1	Koefisien Jarak Euclidean .....	14
2.2.5.2	Koefisien Jarak Korelasi .....	14
2.2.5.3	Koefisien Jarak Kosinus .....	15
2.2.5.4	Koefisien Jarak Canberra .....	15
2.2.6	Transformasi Homografi .....	15
2.2.7	Pembelajaran Mesin .....	16
2.3	Analisis Perbandingan Metode .....	16
BAB III	Metode Penelitian.....	17
3.1	Alat dan Bahan Tugas Akhir .....	17
3.1.1	Alat Tugas Akhir .....	17
3.1.2	Bahan Tugas akhir .....	19
3.2	Tahapan Penelitian .....	19
3.3	Pembuatan Model SQR .....	21
3.3.1	Penentuan Ukuran dan Versi Kode QR .....	21
3.3.2	Penentuan Level Toleransi Kerusakan Kode QR .....	21
3.3.3	Penentuan Ukuran <i>Watermark</i> .....	21
3.3.4	Pembuatan CDP .....	21
3.3.5	Menentukan Jumlah, Jenis, Letak, dan Ukuran Penanda ArUco ...	22
3.3.6	Menempelkan <i>Watermark</i> ke SQR .....	23
3.4	Pembuatan dan Pengolahan <i>Dataset SQR</i> Foto Pertama (Orisinal) .....	24
3.4.1	Pembuatan <i>Batch SQR</i> Foto Pertama (Orisinal).....	24
3.4.2	Pencetakan <i>Batch SQR</i> Foto Pertama (Orisinal) .....	24
3.4.3	Pemotretan <i>Batch SQR</i> Foto Pertama (Orisinal) .....	26
3.4.4	Lokalisasi CDP dari <i>Raw</i> Foto Pertama (Orisinal) .....	26
3.4.4.1	Mengganti Nama Fail.....	26
3.4.4.2	Lokalisasi CDP dari Foto.....	27
3.4.5	Pembuatan Fitur Jarak (Orisinal dengan <i>Template</i> ) .....	29
3.5	Pembuatan dan Pengolahan <i>Dataset SQR</i> Foto Kedua (Palsu) .....	30
3.5.1	Pembuatan Model SQR Foto Kedua (Palsu).....	30
3.5.2	Pembuatan <i>Batch SQR</i> Foto Kedua (Palsu).....	30
3.5.3	Pencetakan <i>Batch SQR</i> Foto Kedua (Palsu) .....	31
3.5.4	Pemotretan <i>Batch SQR</i> Foto Kedua (Palsu) .....	32
3.5.5	Lokalisasi CDP dari <i>Raw</i> Foto Kedua (Palsu) .....	32
3.5.6	Pembuatan Fitur Jarak (Palsu dengan <i>Templat</i> ) .....	32
3.6	Pembuatan dan Pengujian Model Klasifikasi SQR Orisinal dan Palsu .....	32
3.6.1	Pemodelan Klasifikasi Biner menggunakan AutoGluon .....	32

3.7	Metode yang Digunakan.....	32
3.7.1	Pembuatan Model SQR Orisinal .....	32
3.7.1.1	Penentuan Versi Kode QR.....	33
3.7.1.2	Penentuan Level Koreksi Kesalahan .....	33
3.7.1.3	Pembuatan CDP .....	34
3.7.1.4	Pembuatan ArUco <i>Marker</i> di Sekitar CDP .....	34
3.7.1.5	Peletakan <i>Watetemark</i> di Kode QR .....	35
3.7.1.6	Pembuatan <i>Batch</i> SQR Orisinal .....	35
3.7.1.7	Pencetakan dan Pemotretan <i>Dataset</i> SQR Orisinal .....	35
3.7.2	Pengujian Ukuran ArUco <i>Marker</i> .....	35
3.7.3	Mengukur <i>Distance</i> CDP1 dengan CDPT .....	35
3.7.4	Uji Statistik Hasil Pengukuran <i>Distance</i> .....	35
3.7.5	Pembuatan Model SQR Palsu .....	35
3.7.5.1	Mengganti Nama dari Hasil Foto SQR Berdasarkan dengan Data SQR .....	35
3.7.6	Membuat Templat SQR Palsu Berdasarkan Data Nama Foto.....	35
3.7.6.1	Melokalisasi CDP Foto Pertama dan Menempatkannya ke SQR Palsu .....	35
3.7.6.2	Menyimpan SQR Palsu ke Dalam Sebuah Fail.....	35
3.7.6.3	Membuat <i>Batch</i> SQR Palsu .....	35
3.7.6.4	Model Klasifikasi SQR Orisinal dan Palsu .....	35
3.8	Alur Tugas Akhir .....	35
BAB IV	Hasil dan Pembahasan.....	36
4.1	Pembahasan Hasil 1 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)	36
4.2	Pembahasan Hasil 2 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)	36
4.3	Perbandingan Hasil Penelitian dengan Hasil Terdahulu .....	36
BAB V	Tambahan (Opsional) .....	37
BAB VI	Kesimpulan dan Saran .....	38
6.1	Kesimpulan.....	38
6.2	Saran.....	38
DAFTAR	PUSTAKA.....	39
LAMPIRAN	.....	L-1
L.1	Isi Lampiran.....	L-1
L.2	Panduan Latex.....	L-2
L.3	Format Penulisan Referensi .....	L-6
L.4	Contoh Source Code .....	L-10

## **DAFTAR TABEL**

Tabel 2.1	Tabel perbandingan level koreksi kesalahan dengan persentase toleransi kesalahannya.....	12
Tabel 1	Tabel ini .....	L-2
Tabel 2	Contoh tabel panjang .....	L-4

## DAFTAR GAMBAR

Gambar 2.1	Contoh dari CDP a) CDP original hasil generate dari program <i>I</i> b) CDP yang telah terdegradasi kualitasnya akibat dari beberapa kali penyetakan dan pemindaian $\tilde{I}$ .....	5
Gambar 2.2	Perbandingan dari CDP dengan data-matriks: Data-matriks memiliki ukuran unit komponen yang lebih besar, sehingga degradasi informasi tidak berdampak signifikan pada struktur kode [1]	6
Gambar 2.3	Perbandingan dari kode QR asli dan palsu, keduanya menyimpan informasi yang sama dan sama-sama dapat dipindai [2].....	8
Gambar 2.4	SQR diharapkan dapat mendeteksi kode QR palsu pada saat diautentikasi oleh pengguna [2] .....	8
Gambar 2.5	Struktur SQR secara umum [2].....	8
Gambar 2.6	Perangkat seluler melakukan pemindaian menggunakan aplikasi khusus [2] .....	10
Gambar 2.7	Perbandingan 1-D dengan 2-D <i>barcodes</i> .....	11
Gambar 2.8	Struktur modul pada kode QR dua dimensi .....	11
Gambar 2.9	Jumlah modul berdasarkan versi kode QR.....	12
Gambar 2.10	Salah satu contoh CDP .....	13
Gambar 3.1	Diagram alir tahapan penelitian .....	20
Gambar 3.2	Sampel 50 penanda ArUco dengan ukuran modul 4x4 piksel ....	23
Gambar 3.3	Sampel 50 penanda ArUco dengan ukuran modul 6x6 piksel ....	23
Gambar 3.4	<i>Batch</i> SQR orisinal (foto pertama) dalam piksel .....	24
Gambar 3.5	Konfigurasi saat melakukan pencetakan menggunakan Adobe Photoshop .....	25
Gambar 3.6	Konfigurasi menambahkan ukuran kertas A3+ dan cetak menjadi PDF .....	25
Gambar 3.7	Nama fail sebelum perubahan .....	26
Gambar 3.8	Nama fail sebelum perubahan .....	26
Gambar 3.9	xy_list yang menyimpan koordinat titik tengah penanda ArUco dari id 0 s.d. 7 .....	27
Gambar 3.10	Hasil plot koordinat titik tengah penanda ArUco pada xy_list ke gambar hasil foto.....	28
Gambar 3.11	<i>List</i> koordinat tujuan transformasi homografi .....	28
Gambar 3.12	SQR awal dan setelah dilakukan transformasi homografi .....	29
Gambar 3.13	Perbandingan CDP <i>template</i> dengan CDP hasil lokalisasi foto pertama (orisinal) .....	30
Gambar 3.14	Hasil <i>dataset</i> CDP foto pertama (orisinal) yang sudah dilokalisasi dan diganti nama failnya.....	31
Gambar 3.15	<i>Dataframe</i> fitur jarak <i>dataset</i> foto pertama (orisinal) .....	31
Gambar 3.16	SQR diharapkan dapat mendeteksi kode QR palsu pada saat diautentikasi oleh pengguna [2] .....	34
Gambar 1	Contoh gambar. .....	L-2

## **DAFTAR SINGKATAN**

### **[SAMPLE]**

$b$	= bias
$K(x_i, x_j)$	= fungsi kernel
$y$	= kelas keluaran
$C$	= parameter untuk mengendalikan besarnya pertukaran antara penalti variabel slack dengan ukuran margin
$L_D$	= persamaan Lagrange dual
$L_P$	= persamaan Lagrange primal
$\mathbf{w}$	= vektor bobot
$\mathbf{x}$	= vektor masukan
QR	= Quick Response
SQR	= Secure Quick Response Code
CDP	= Copy Detection Pattern
ANFIS	= Adaptive Network Fuzzy Inference System
ANSI	= American National Standards Institute
DAG	= Directed Acyclic Graph
DDAG	= Decision Directed Acyclic Graph
HIS	= Hue Saturation Intensity
QP	= Quadratic Programming
RBF	= Radial Basis Function
RGB	= Red Green Blue
SV	= Support Vector
SVM	= Support Vector Machines

## **INTISARI**

Intisari ditulis menggunakan bahasa Indonesia dengan jarak antar baris 1 spasi dan maksimal 1 halaman. Intisari sekurang-kurangnya berisi tentang latar belakang dan tujuan penelitian, metodologi yang digunakan, hasil penelitian, kesimpulan dan implikasi, dan Kata kunci yang berhubungan dengan penelitian.

Kata Kunci ditulis maksimal 5 kata yang paling berhubungan dengan isi skripsi. Silakan mengacu pada ACM / IEEE *Computing classification* jika Anda adalah mahasiswa Sarjana TI <http://www.acm.org/about/class/> atau mengacu kepada IEEE keywords [http://www.ieee.org/documents/taxonomy\\_v101.pdf](http://www.ieee.org/documents/taxonomy_v101.pdf) jika Anda berasal dari Prodi Sarjana TE.

Kata kunci : Kata kunci 1, Kata kunci 2, Kata kunci 3, Kata kunci 4, Kata kunci 5

## **ABSTRACT**

*Abstract ditulis italic (miring) menggunakan bahasa Inggris dengan jarak antar baris 1 spasi dan maksimal 1 halaman. Abstract adalah versi Bahasa Inggris dari intisari. Abstract dapat ditulis dalam beberapa paragraf. Baris pertama paragraph harus menjorok ke dalam sekitar 1 cm. Tidak dsarankan menggunakan mesin penerjemah melainkan tulis ulang.*

**Keywords :** Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pembajakan produk atau yang dikenal juga sebagai tindakan pemalsuan, merupakan suatu tindakan ilegal yang dilakukan dengan tujuan memperoleh keuntungan dengan cara meniru atau menyalin produk asli yang sudah dipatenkan atau memiliki hak cipta. Pembajakan produk semakin marak di era digital dan globalisasi, seiring dengan perkembangan teknologi. Di era digital, pembajakan produk semakin mudah dilakukan dengan memanfaatkan internet dan teknologi digital. Sementara itu, globalisasi mempermudah transportasi dan distribusi produk palsu dari satu negara ke negara lain. Pembajakan produk juga menyebabkan kerugian ekonomi yang signifikan bagi produsen dan pemilik hak merek, serta dapat membahayakan keselamatan konsumen. Hal ini terjadi karena pembajakan produk dapat merusak citra perusahaan, mengurangi pendapatan, serta merugikan konsumen yang membeli produk palsu yang seringkali berkualitas rendah dan dapat membahayakan diri secara langsung. [3]

Teknologi percetakan dan pemindai digital telah mengalami perkembangan yang pesat selama beberapa dekade terakhir. Namun, kemajuan ini tidak hanya digunakan untuk kegiatan positif juga dimanfaatkan oleh pelaku pembajakan untuk memproduksi produk-produk bajakan. Dengan kemampuan teknologi percetakan dan pemindai yang semakin canggih, pembajakan produk menjadi lebih mudah, lebih cepat, dan lebih murah. [4]

Printer 2D beresolusi tinggi dan Printer 3D, memungkinkan pembuat produk bajakan untuk membuat produk-produk dengan kualitas hampir sama dengan produk asli. Pemindai 3D juga pembuat produk bajakan untuk menyalin produk asli hingga detail terkecil dengan cepat dan mudah. Selain itu, teknologi digital seperti desain grafis dan software pemodelan juga memudahkan pelaku pembajakan dalam membuat desain dan cetakan produk tanpa harus membeli hak cipta atau paten produk tersebut. [5]

Kerugian dari praktik ilegal dan tidak etis ini diperkirakan mencapai \$4,2 triliun per tahun, dan terus meningkat dengan sangat cepat. Beberapa cara yang dapat dilakukan untuk melawan pembajakan, antara lain melalui, komunikasi, pemerintah, hukum, kontak langsung, pelabelan, pemasaran proaktif, dan mempromosikan perlawanannya terhadap pembajakan. Oleh karena itu, upaya preventif perlu dilakukan untuk meminimalisir dan melakukan perlawanannya terhadap pembajakan produk. [6]

## **1.2 Rumusan Masalah**

Dari masalah yang telah dijelaskan pada bagian latar belakang, yaitu semakin maraknya pembajakan dan pemalsuan produk seiring dengan berkembangnya teknologi perangkat pemindai digital dan juga teknologi percetakan, penulis mencoba untuk menerapkan CDP ke dalam SQR yang akan digunakan untuk mendeteksi pemalsuan produk. SQR yang didesain penulis dilengkapi dengan *watermark* yang di dalamnya terdapat CDP dan delapan penanda ArUco. *Watermark* tersebut nantinya akan digunakan untuk proses autentikasi, membedakan SQR orisinal dan palsu.

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah:

1. Membuat model SQR yang dilengkapi *watermark* (CDP dan delapan *marker*) yang dapat diautentikasi oleh model.
2. Mengetahui nilai parameter P&S dalam pembuatan SQR (konfigurasi kamera dan penyetakan).
3. Mengetahui hasil deteksi objek CDP dalam SQR menggunakan delapan ArUco *marker* yang diletakkan di sekitar CDP.
4. Membuat model untuk melakukan autentikasi SQR orisinal dan palsu.
5. Mengetahui karakteristik CDP 2 dan 4 level.
6. Mengetahui performa model dalam mengautentikasi CDP dengan dua dan empat kuantisasi *grayscale*.
7. Mengetahui efektivitas CDP untuk mendeteksi SQR orisinal dan palsu.

## **1.4 Batasan Penelitian**

Beberapa batasan yang penulis gunakan dalam penelitian ini antara lain:

1. Objek Penelitian: Pengembangan dan analisis pembuatan SQR dengan *watermark* (CDP 2 dan 4 level disertai delapan penanda ArUco)
2. Metode Penelitian: Pengembangan SQR dilakukan dengan eksperimen yang mencakup pembuatan model SQR, pembuatan dataset SQR orisinal dan palsu, serta pembuatan model untuk membedakan SQR orisinal dan palsu. Hasil dari penelitian ini adalah mengetahui apakah SQR yang dilengkapi dengan CDP 2 dan 4 level dapat digunakan oleh model autentikasi untuk mengklasifikasikan SQR orisinal dan palsu.
3. Waktu dan Tempat Penelitian:
  - Waktu penelitian: Agustus 2022 s.d. April 2023

- Tempat penelitian: Ruang rispro, laboratorium informatika, laboratorium jaringan dan komputer, dan rumah penulis.
4. Populasi dan Sampel: Populasi penelitian berupa seluruh *dataset* SQR yang dibuat dan digunakan dalam penelitian. Sampel penelitian berupa sebagian *dataset* SQR yang digunakan untuk penentuan parameter dan pembuatan model klasifikasi biner.
  5. Variabel: Variabel bebas meliputi *dataset* SQR 2 dan 4 level, sedangkan variabel terikatnya adalah hasil lokalisasi objek CDP, parameter model SQR, parameter P&S, dan hasil klasifikasi biner model untuk mengklasifikasikan SQR orisinal atau palsu.
  6. Hipotesis:
    - CDP akan terdegradasi dari proses P&S.
    - CDP dapat digunakan untuk membedakan SQR orisinal dan palsu melalui proses autentikasi.
    - Proses autentikasi untuk mengklasifikasikan SQR orisinal dan palsu dapat dibuat menggunakan AutoML.
    - Delapan penanda ArUco yang diletakkan di dalam *watermark* pada SQR dapat membantu melokalisasi objek CDP.
    - Kualitas CDP 4 level lebih sulit terdegradasi dibandingkan CDP 2 level.
  7. Keterbatasan Penelitian:
    - *Printer* yang digunakan untuk mencetak *dataset* SQR seragam.
    - Pemotretan *dataset* SQR dilakukan dalam kondisi pencahayaan yang baik, berasal dari *flash smartphone*.
    - Kamera, konfigurasi kamera, sudut dan kondisi pemotretan *dataset* adalah tetap.

## 1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini, pengembang SQR dua dimensi dapat menggunakan model SQR dan model autentikasi yang memiliki hasil klasifikasi biner terbaik (dapat membedakan SQR orisinal dan palsu). Selain itu, peletakan ArUco marker diharapkan mampu mendeteksi CDP dalam SQR yang nantinya akan diambil fiturnya dengan lebih cepat dan akurat. Bagi peneliti, penelitian ini dapat menambah wawasan, ilmu dan pengetahuan dalam pembuatan tulisan ilmiah, khususnya pada topik keamanan digital, pengolahan citra gambar, dan pembelajaran mesin. Bagi pelaku bisnis, penerapan SQR dapat membantu mereka dalam melindungi produk mereka dari pembajakan.

## **1.6 Sistematika Penulisan**

### **BAB I : PENDAHULUAN**

Pada bab ini dijelaskan latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI**

Pada bab ini dijelaskan teori-teori dan penelitian terdahulu yang digunakan sebagai acuan dan dasar dalam penelitian.

### **BAB III : METODOLOGI PENELITIAN**

Pada bab ini dijelaskan metode yang digunakan dalam penelitian meliputi langkah kerja, pertanyaan penelitian, alat dan bahan, serta tahapan dan alur penelitian.

### **BAB IV : HASIL DAN PEMBAHASAN**

Pada bab ini dijelaskan hasil penelitian dan pembahasannya.

### **BAB V : KESIMPULAN DAN SARAN**

Pada bab ini ditulis kesimpulan akhir dari penelitian dan saran untuk pengembangan penelitian selanjutnya.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

##### 2.1.1 Apakah Pola Deteksi Duplikat (CDP) dapat Disalin

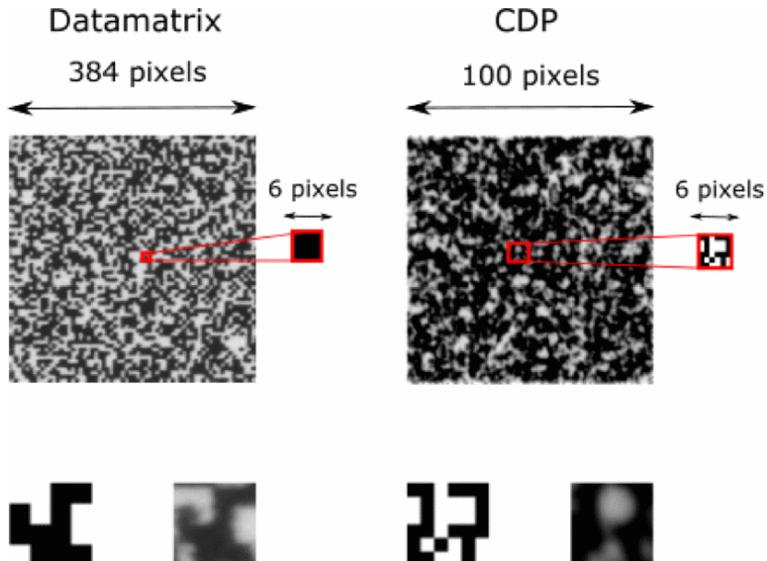
CDP dinilai dapat digunakan untuk mendeteksi pemalsuan, sehingga akhir-akhir ini mendapatkan banyak perhatian dari akademisi dan industri. Tingkat keamanan CDP dalam mendeteksi serangan pemalsuan yang canggih telah dipelajari secara teoritis dan praktis dalam beberapa penelitian, namun hasilnya masih belum sepenuhnya meyakinkan [1]. Kontribusi utama dari penelitian ini adalah untuk menyajikan *dataset* secara publik dan berbagai jenis contoh penyerangan terdapat CDP tersebut, sehingga kinerja CDP terhadap beberapa penyerangan dapat diketahui. *Dataset* CDP tersebut terdiri dari lebih dari 27.500 gambar CDP dan merupakan set data CDP terbesar hingga saat ini [1]. Kontribusi selanjutnya adalah meneliti tentang kinerja detektor CDP dalam mendeteksi CDP asli ataupun salinan. Verifikasi dari CDP dapat dilakukan dengan perangkat seluler, tanpa harus menggunakan pemindai khusus [7], [8]. Dengan dataset dan detektor yang dibuat, peneliti sebelumnya ingin menguji validitas hipotesis bahwa CDP yang dicetak berulang kali akan terdegradasi kualitasnya dan dapat dideteksi sebagai CDP palsu [1]. Kontibusi terakhir dari penelitian ini adalah meneliti beberapa metode yang dapat dilakukan untuk meningkatkan performa klasifikasi dari model.



Gambar 2.1. Contoh dari CDP a) CDP original hasil generate dari program *I* b) CDP yang telah terdegradasi kualitasnya akibat dari beberapa kali penyetakan dan pemindaian  $\tilde{I}$  [1]

##### 2.1.1.1 Prinsip Degradasi Informasi

Prinsip dari deteksi CDP palsu dilakukan berdasarkan hilangnya informasi, yang mana muncul dari proses pemindaian dan penyetakan [9]. Proses *Print-and-Scan* (P&S) merupakan proses stokastik (mempunyai unsur peluang atau kebolehjadian) [10], yang menyebabkan perubahan struktur dan kualitas gambar pada CDP, seperti yang ditampilkan pada Gambar 2.1 *Noise* yang dihasilkan dari P&S CDP sulit untuk dikarakterisasi [11] karena setiap printer dan pemindai memiliki karakteristiknya sendiri.



Gambar 2.2. Perbandingan dari CDP dengan data-matriks: Data-matriks memiliki ukuran unit komponen yang lebih besar, sehingga degradasi informasi tidak berdampak signifikan pada struktur kode [1]

CDP sering dibandingkan dengan kode batang dua dimensi seperti data-matriks karena kemiripan visualnya. Namun, seperti yang diilustrasikan pada Gambar 2.2, unit elemen data-matriks jauh lebih besar dibandingkan unit elemen CDP. Oleh karena itu, prinsip kehilangan atau degradasi informasi tidak berdampak pada struktur data-matriks. Sebagai contoh, korelasi antara data-matriks digital *template* hasil *generate* dengan versi rusak P&S bisa lebih dari 0,9, sedangkan korelasi antara CDP digital *template* hasil *generate* dengan versi rusak P&S-nya hanya sekitar 0,45 – 0,55 tergantung pada pemindai dan penyetaknya. Oleh karena itu, ukuran dari unit elemen pola CDP, yaitu  $uxu$  piksel seperti yang ditampilkan pada Gambar 2.2, ukuran pola keseluruhan juga sangat mempengaruhi proses otentikasi dan kemampuan pemalsu dalam mereproduksi pola. Pada praktiknya, ukuran unit elemen pada CDP adalah  $1x1$  piksel atau  $2x2$  piksel agar bisa memanfaatkan prinsip kehilangan atau degradasi informasi secara maksimal.

### 2.1.1.2 Definisi Teoritis dari Sistem Autentikasi CDP

Autentikasi dari CDP terdiri dari dua langkah utama. Langkah pertama adalah langkah registrasi di mana pola dihasilkan kemudian dicetak dengan penyetak untuk menghasilkan CDP asli. Langkah kedua adalah verifikasi CDP, menggunakan sebuah perangkat pemindai yang telah terotentikasi (perangkat selular dengan kamera), CDP dipindai dan dilewatkan ke tes autentikasi (dengan parameter-parameter peminadaian tertentu). Jika tes tersebut positif, item dianggap autentik.

Upaya penyerangan yang paling sering terjadi oleh pembajak adalah sebagai berikut: Pembajak melakukan pemindaian CDP pada sebuah item menggunakan pemindai beresolusi tinggi, mengestimasi pola asli dari CDP, kemudian mencetak pola yang te-

lah diestimasi menggunakan penyetak beresolusi tinggi. Pada skenario ini  $I$  merupakan CDP digital hasil *generate* dari *template*, kemudian  $\Pi(I)$  merupakan hasil *generate* CDP *template* yang dicetak, dengan  $\Pi(\cdot)$  noise yang dihasilkan dalam proses penyetakan menggunakan penyetak yang telah terautentikasi. Selanjutnya, proses autentikasi dapat dirumuskan sebagai uji hipotesis berikut ini:

$$\begin{aligned}\mathcal{H}_0 : \tilde{I} &\sim \Sigma(\Pi(I)), \\ \mathcal{H}_1 : \tilde{I} &\not\sim \Sigma(\Pi(I)),\end{aligned}\tag{2-1}$$

di mana  $\tilde{I}$  adalah gambar CDP *grayscale* yang diterima oleh pusat autentikasi.  $\tilde{I}$  dapat berupa CDP asli (i.e.  $\Sigma(\Pi(I))$ ) atau CDP palsu (i.e.  $\Sigma(\Pi'(\hat{I}))$ ). Metriks yang digunakan untuk membandingkan CDP asli dengan palsu adalah koefisien jarak ataupun korelasi [12].

### 2.1.1.3 Komponen dari Detektor

Hasil menunjukkan bahwa autentikasi menggunakan CDP *raw grayscale* lebih efisien dibandingkan dengan CDP yang telah di-*thresholding* [10]. Kemudian, secara umum, ada beberapa langkah yang dilakukan untuk melakukan autentikasi CDP, antara lain:

- Melakukan *resizing* pada *template* CDP menggunakan faktor skala tertentu.
- Menggunakan teknik pencocokan *template* dengan mengambil sub-bagian dari CDP.
- Menggunakan *high pass filtering* (seperti *unsharp masking*) sebelum melakukannya korelasi.

### 2.1.2 Detekti Pembajakan menggunakan SQR

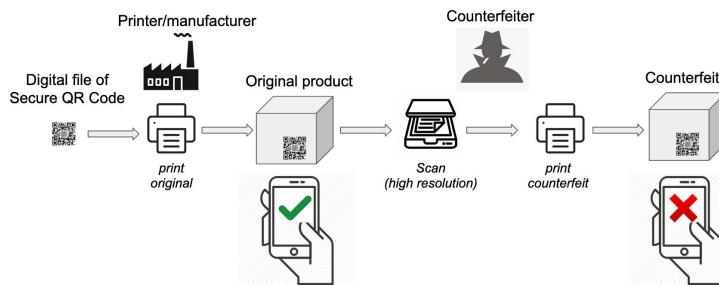
Pendekatan keamanan tradisional pada produk sebelumnya sudah diimplementasikan melalui *taggant*, hologram, dan tinta keamanan. Beberapa metode tersebut memang mudah diimplementasikan, mudah diverifikasi, dan murah. Namun, dalam sebuah Kode QR metode-metode tersebut belum dapat diimplementasikan. Produk yang ditempel pada kode QR palsu biasanya masih dapat dipindai dan mengembalikan keluaran sama dengan produk asli. Penelitian yang dilakukan oleh Justin Picard, Paul Landry, dan Michael Bolay ini membahas tentang bagaimana mengimplementasikan CDP ke dalam SQR untuk mendeteksi pembajakan [2].

Pada Gambar 2.3 terlihat bahwa sangat sulit untuk membedakan antara kode QR asli dan replika, apabila jika tidak ada pembandingnya. Skenario yang dapat dilakukan



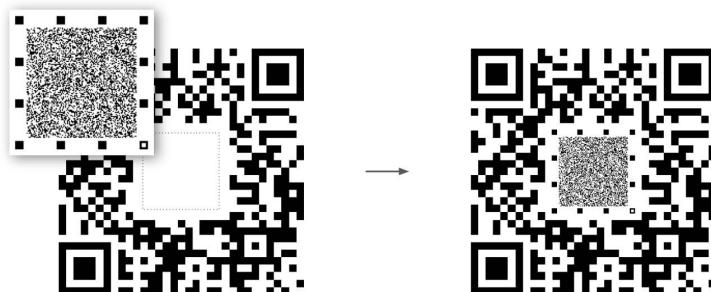
Gambar 2.3. Perbandingan dari kode QR asli dan palsu, keduanya menyimpan informasi yang sama dan sama-sama dapat dipindai [2]

oleh pembajak dalam menerbitkan kode QR palsu dapat dilihat pada Gambar 2.4. Kode QR palsu dipindai dengan perangkat beresolusi tinggi, kemudian dicetak ulang. Harapannya dengan CDP yang diletakkan pada kode QR, kode QR palsu yang dipindai untuk autentikasi dapat terdeteksi sebagai kode QR palsu, lihat pada Gambar 2.4.



Gambar 2.4. SQR diharapkan dapat mendeteksi kode QR palsu pada saat diautentikasi oleh pengguna [2]

#### 2.1.2.1 Struktur dari SQR



Gambar 2.5. Struktur SQR secara umum [2]

Secara umum, CDP akan diletakkan di tengah-tengah dari kode QR. CDP nantinya akan digunakan untuk proses autentikasi. Di sekitar CDP diletakkan beberapa *markers* untuk memudahkan program dalam mendeteksi dan mendapatkan objek CDP. CDP

mungkin ditempelkan di tengah-tengah kode QR karena adanya fitur koreksi kesalahan dalam kode QR. Ada empat jenis koreksi kesalahan pada kode QR: L, M, Q, dan H, yang secara teoritis memiliki kemampuan untuk memulihkan informasi yang hilang sebesar 7%, 15%, 25%, dan 30% kerusakan pada kode QR. Pada penelitian ini, area yang dirusak untuk meletakkan CDP adalah sebesar 1/9 dari ukuran awal kode QR. Aplikasi yang digunakan untuk melakukan autentikasi pada perangkat pemindai (perangkat seluler) hanya mengambil objek CDP saja, hal tersebut memiliki beberapa keuntungan, area yang digunakan untuk autentikasi menjadi lebih spesifik, sehingga kecepatan deteksi akan lebih cepat, fokus dari kamera juga relatif akan lebih baik karena mengambil objek yang lebih kecil dan terfokus, selain itu karena sistem autentikasi berada di *remote server* yang mana data akan dikirim dari perangkat seluler pemindai, maka semakin kecil area yang dikirimkan, semakin hemat *bandwidth* yang digunakan.

### 2.1.2.2 Pembuatan dan Pencetakan SQR

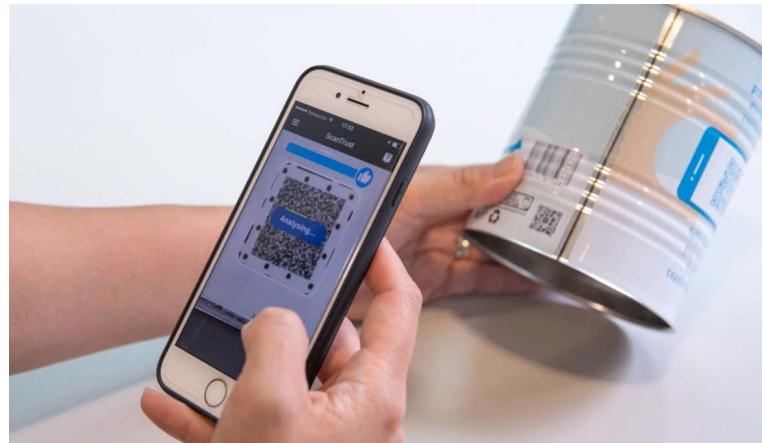
CDP yang digunakan dalam penelitian ini menggunakan dua kuantisasi *grayscale* atau biner. Resolusi mesin *printer* yang digunakan adalah 812,8 ppi dengan merek HP Indigo. CDP di-*generate* menggunakan *pseudo-random number generator*, menggunakan *seed* tertentu. CDP dapat dibuat unik untuk setiap kode QR ataupun sama pada sekelompok kode QR tertentu. Dalam melakukan penyetakan dan pemindaian SQR, perangkat pencetak dan pemindai akan diverifikasi terlebih dahulu dengan konfigurasi dan parameter tertentu untuk menjamin kualitas dari pencetakan.

### 2.1.2.3 Autentikasi SQR

Perangkat pemindai QR biasa sudah pasti dapat melakukan *decode* informasi dari kode QR, namun untuk melakukan autentikasi terhadap CDP, tentunya diperlukan aplikasi khusus. Aplikasi khusus ini berjalan di perangkat seluler, secara umum proses pemindaian yang dilakukan oleh aplikasi khusus tersebut adalah sebagai berikut:

- Aplikasi akan melakukan pemindaian per-*frame* dari kamera hingga mendapatkan kode QR.
- Kode QR akan di-*decode* untuk mengekstrak informasi dalam kode QR.
- Indeks kualitas pemindaian akan dikalkulasi berdasarkan *frame* yang didapatkan.
- Jika indeks kualitas pemindaian dinilai cukup tinggi, area CDP akan didekripsi, dipotong, dan dikirim ke *remote server* bersamaan dengan informasi dari kode QR yang telah ter-*decode* untuk autentikasi.

Setelah *server* mendapatkan CDP yang telah dipotong beserta informasi data kode QR, autentikasi yang dilakukan di *server* adalah sebagai berikut:



Gambar 2.6. Perangkat seluler melakukan pemindaian menggunakan aplikasi khusus [2]

- *Identifier* unik akan diekstrak dari informasi kode QR yang didapatkan yang mana dibutuhkan sebagai parameter autentikasi.
- Template CDP digital akan di-*generate*.
- Matriks similaritas akan dikalkulasi dari perbandingan antara CDP yang dikirimkan dari hasil pemindaian dengan CDP *template* yang di-*generate*.
- Penyesuaian lain seperti, ketajaman gambar, filter akan dilakukan untuk memperoleh hasil terbaik.
- Normalisasi nilai akan dilakukan.
- Keluaran berupa CDP "asli", "palsu", atau "pemindaian buruk" akan dikeluarkan oleh *server* (pemindaian buruk bisa disebabkan oleh hasil gambar yang blur).

### 2.1.3 Autentikasi Digital menggunakan CDP

### 2.1.4 Pencetakan Variasi CDP

### 2.1.5 Autentikasi CDP menggunakan Perangkat Seluler

## 2.2 Dasar Teori

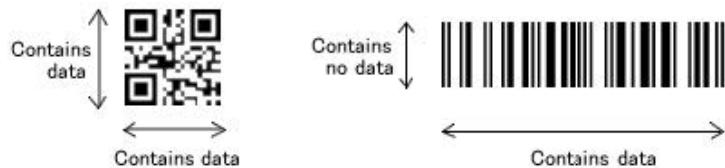
### 2.2.1 Kode QR

Kode QR (Quick Response) adalah sebuah kode matriks dua dimensi (2-D) yang dapat dibaca oleh komputer. Kode QR dua dimensi dapat menyimpan data yang lebih banyak dibandingkan dengan kode satu dimensi (barcodes) dengan ruang yang lebih kecil. Selain itu, kode QR memiliki fitur koreksi kesalahan pembacaan dan beberapa fitur unik lainnya [13].

Seperti bahasa tertulis lainnya, kode batang atau *barcode* merupakan representasi visual dari informasi. Namun, berbeda dengan bahasa yang dapat dibaca oleh manusia,

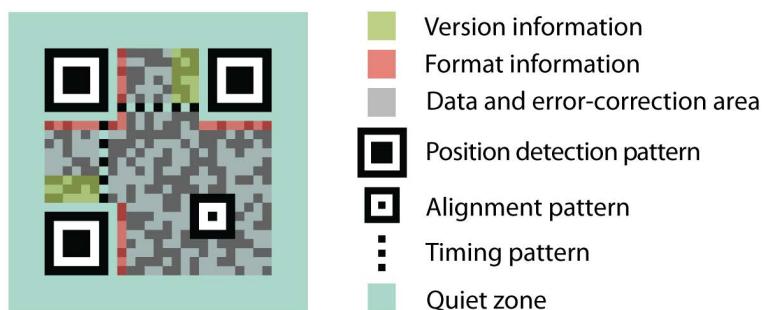
kode batang dirancang untuk dibaca dan dipahami oleh komputer atau mesin. Menggunakan sistem penglihatan dari mesin berupa pemindai laser optik ataupun kamera dan perangkat lunak yang dapat menginterpretasikan kode batang. Aturan bagaimana *barcode* dikonstruksikan disebut sebagai *grammar*, sedangkan set karakter yang digunakan (alfabet) disebut *symbology* [13].

### 2.2.1.1 Bagaimana Kode QR Bekerja



Gambar 2.7. Perbandingan 1-D dengan 2-D *barcodes*

Tidak seperti kode batang satu dimensi, kode QR adalah matriks 2-D yang menyimpan informasi dalam tiap modul di baris dan kolomnya yang memiliki gelap dan terang tertentu.



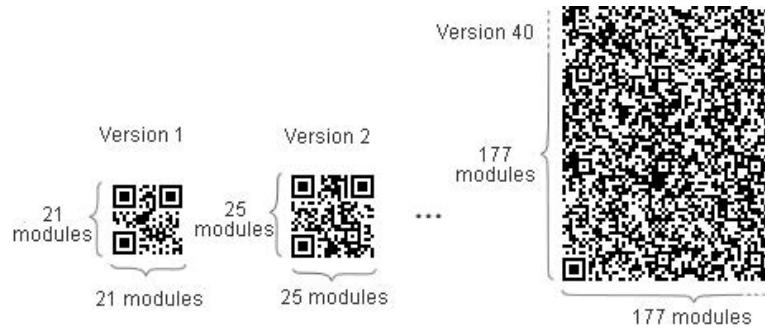
Gambar 2.8. Struktur modul pada kode QR dua dimensi

Setiap modul dalam kode QR memiliki fungsi-fungsi tertentu. Beberapa modul berisi tentang informasi yang tersimpan dalam kode QR itu sendiri, dengan lainnya dibagi menjadi beberapa grup berdasarkan fungsinya. Untuk memastikan kode QR dapat dipindai dari berbagai sisi, ada tiga modul *position detection pattern* yang terletak di ketiga sudut yang memungkinkan kode QR untuk dipindai dari 360°.

### 2.2.1.2 Versi Kode QR

Kode QR dapat di-*generate* dari 40 versi yang berbeda, dari yang berukuran  $21 \times 21$  modul (versi 1) hingga  $177 \times 177$  modul (versi 40).

Setiap kenaikan satu versi, maka akan ada penambahan 4 modul, sehingga dapat memuat data atau informasi yang lebih banyak. Jumlah maksimum data yang dapat disimpan tergantung pada versi kode QR, tipe karakter, dan besar toleransi kesalahan.



Gambar 2.9. Jumlah modul berdasarkan versi kode QR

### 2.2.1.3 Koreksi Kesalahan Kode QR

Koreksi kesalahan kode QR mengimplementasikan *Reed-Solomon codes*, yang mana merupakan salah satu metode koreksi kesalahan matematis yang banyak digunakan. Hal ini memungkinkan kode QR tetap dapat dibaca walaupun dalam kondisi kotor ataupun rusak dengan batasan tertentu. Ada empat tipe koreksi kesalahan standar yang ada dalam kode QR. Semakin tinggi level koreksi kesalahan, semakin besar toleransi terhadap kerusakan kode QR, namun semakin besar juga versi kode QR-nya.

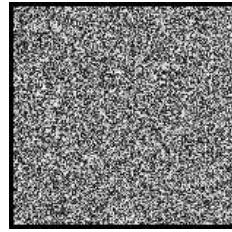
Tabel 2.1. Tabel perbandingan level koreksi kesalahan dengan persentase toleransi kesalahan

Level Koreksi Kesalahan	Besar Toleransi Kesalahan
L	7%
M	15%
Q	25%
H	30%

Dalam memilih level koreksi kesalahan, sebaiknya disesuaikan dengan kondisi lingkungan di mana kode QR tersebut digunakan. Misalnya dalam kondisi lingkungan yang bersih, level L (7%) dapat digunakan. Secara umum, level yang paling sering digunakan adalah level M (15%).

### 2.2.2 Deteksi Pola Duplikat (CDP)

Pola deteksi duplikat (CDP) adalah sebuah gambar berpola dengan entropi tinggi yang dibuat menggunakan kode rahasia (*secret key*). CDP memanfaatkan konsep dari "*information loss principle*" dari proses P&S pada dokumen. Biasanya, CDP digunakan di dalam gambar digital yang dicetak ataupun langsung ke dalam dokumen digitalnya. CDP tidak didesain untuk dideteksi menggunakan mata telanjang. Namun, CDP dapat bekerja secara maksimal pada pendekripsi otomatis pada gambar yang dipindai. CDP akan sangat bermanfaat saat digunakan dalam memverifikasi dokumen dalam jumlah besar [9], [14], [].



Gambar 2.10. Salah satu contoh CDP

CDP telah dicoba untuk dicetak menggunakan berbagai variasi *printers*: *Printer* kantor seperti *inkjet* dan *laser*, *printer offset* dan *digital offset*, dan juga *printer* termal. Hasil percobaan dari pencetakan menggunakan beberapa jenis *printer* tadi, semua CDP salinan dapat dibedakan dengan CDP asli dengan margin yang cukup nyaman [9], [14].

CDP sebaiknya tidak dianggap sebagai pesaing untuk perangkat keamanan optik lainnya, namun dijadikan sebagai alternatif yang lebih murah pada kasus-kasus tertentu. Dengan memanfaatkan konsep degradasi gambar dan informasi yang tidak terancam oleh perkembangan perangkat pemindai dan pencetak digital, CDP menjadi alternatif yang paling murah dalam memroteksi dokumen [9], [14], [15].

### 2.2.3 Lokalisasi Objek dengan Pengenalan Pola

Lokalisasi objek adalah proses mengidentifikasi posisi dan orientasi objek atau pola tertentu pada sebuah gambar menggunakan teknik pengolahan citra dan visi komputer. Proses ini melibatkan deteksi objek atau pola dalam gambar serta mengestimasi lokasi dan orientasi yang tepat relatif terhadap kamera [16], [17].

Berbagai teknik telah diusulkan untuk melakukan lokalisasi objek atau gambar dengan pengenalan pola, termasuk deteksi fitur, pencocokan *template*, dan metode berbasis pembelajaran mesin. Teknik-teknik ini telah digunakan dalam berbagai aplikasi, seperti *augmented reality*, robotika, dan pelacakan objek.

### 2.2.4 ArUco Marker

ArUco *marker* adalah jenis *marker* fidusial yang biasa digunakan untuk estimasi pose kamera dan pelacakan dalam pengolahan citra dan visi komputer. ArUco *marker* terdiri dari kisi-kisi kotak hitam dan putih dengan pola unik yang mudah dideteksi dan dikenali oleh algoritma visi komputer. ArUco *marker* banyak digunakan dalam aplikasi robotika, realitas tambahan, dan pelacakan objek karena kesimpelannya, akurasi deteksi yang tinggi, dan biaya komputasi yang rendah.

*Marker* fidusial adalah objek berpola yang digunakan ke dalam sebuah gambar atau tempat tertentu untuk memudahkan sistem visi komputer menentukan posisi dan orientasi objek atau *scene* dengan akurasi yang tinggi. *Marker* fidusial biasanya didesain

dengan pola atau bentuk yang unik dan mudah dikenali, sehingga dapat dideteksi dan dilacak oleh algoritme pendekripsi objek visi komputer, yang memungkinkan estimasi pose dan pelacakan yang akurat. *Marker* fidusial banyak digunakan dalam berbagai aplikasi seperti *augmented reality*, robotika, dan pendekripsi objek.

## 2.2.5 Koefisien Jarak

Koefisien jarak merupakan ukuran atau besaran yang menggambarkan seberapa dekat atau jauh dua objek dalam ruang atau dimensi tertentu. Koefisien jarak dapat digunakan untuk mengukur kesamaan atau perbedaan antara dua objek yang diamati. Ada beberapa koefisien jarak yang sering digunakan, antara lain:

### 2.2.5.1 Koefisien Jarak Euclidean

Jarak Euclidean adalah ukuran jarak yang paling umum digunakan dalam matematika dan ilmu komputer untuk mengukur jarak antara dua titik dalam ruang Euclidean n-dimensi. Jarak Euclidean dihitung sebagai akar kuadrat dari jumlah kuadrat perbedaan koordinat antara dua titik. Secara formal, jarak Euclidean antara dua vektor  $\mathbf{u}$  dan  $\mathbf{v}$  dalam ruang Euclidean n-dimensi didefinisikan sebagai:

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^n (u_i - v_i)^2} \quad (2-2)$$

di mana  $n$  adalah jumlah dimensi dalam *vector space*, dan  $u_i$  dan  $v_i$  merepresentasikan komponen ke- $i$  pada vektor.

### 2.2.5.2 Koefisien Jarak Korelasi

Koefisien jarak korelasi adalah suatu ukuran kemiripan atau perbedaan antara dua vektor, berdasarkan korelasi antara komponen-komponennya. Nilai dari jarak korelasi dinormalisasi antara 0 dan 1, di mana 0 menunjukkan korelasi positif sempurna sedangkan 1 menunjukkan korelasi negatif sempurna.

Untuk menghitung jarak korelasi antara dua vektor  $u$  dan  $v$ , dapat dituliskan dengan:

$$d = 1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{\|(u - \bar{u})\|_2 \|(v - \bar{v})\|_2} \quad (2-3)$$

di mana  $d$  adalah jarak korelasi antara  $u$  dan  $v$ ,  $\bar{v}$  adalah rata-rata elemen dari vektor  $v$ , dan  $x \cdot y$  adalah *dot product* dari  $x$  dan  $y$ . Jarak korelasi sering digunakan dalam algoritma pengelompokan dan klasifikasi untuk mengukur perbedaan antara sampel, terutama ketika vektor memiliki banyak komponen dan data sangat berkorelasi.

### 2.2.5.3 Koefisien Jarak Kosinus

Koefisien jarak kosinus adalah sebuah metode untuk mengukur kemiripan antara dua vektor dalam ruang n-dimensi [18], [19]. Metode ini mengukur sudut antara dua vektor dan menghasilkan nilai berkisar antara 0 dan 1, di mana 0 menunjukkan bahwa vektor tersebut saling tegak lurus, sedangkan 1 menunjukkan bahwa vektor tersebut saling sejajar. Semakin kecil nilai koefisien jarak kosinus, semakin mirip kedua vektor tersebut. Jarak kosinus antara dua vektor  $u$  dan  $v$  dapat dituliskan sebagai berikut:

$$1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2} \quad (2-4)$$

di mana  $u \cdot v$  adalah hasil *dot product* antara vektor  $u$  dan  $v$ , sedangkan  $\|u\|_2$  dan  $\|v\|_2$  adalah panjang dari vektor  $u$  dan  $v$ .

### 2.2.5.4 Koefisien Jarak Canberra

Jarak Canberra adalah salah satu metode mengukur jarak antara dua vektor dalam ruang n-dimensi. Metode ini mengevaluasi perbedaan proporsional antara nilai-nilai pada setiap elemen vektor. Metode ini sering digunakan dalam analisis data untuk mengukur kemiripan antara dua set data numerik [20], [21]. Secara formal, perhitungan jarak canberra antara dua vektor  $u$  dan  $v$  adalah:

$$d(u, v) = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|} \quad (2-5)$$

### 2.2.6 Transformasi Homografi

Transformasi homografi adalah sebuah transformasi geometri pada ruang n-dimensi yang memetakan setiap titik pada bidang ke titik yang sesuai pada bidang lainnya, dengan menerapkan konsep persamaan linier homogen. Transformasi homografi dapat mengubah ukuran, rotasi, dan persepektif dari gambar atau objek pada bidang ruang n-dimensi.

Transformasi homografi biasanya dilakukan dalam ruang koordinat *homogeneous*, yang merupakan ruang  $n+1$  dimensi dengan koordinat homogen yang memungkinkan dilakukannya transformasi perspektif. Dalam ruang koordinat *homogeneous*, sebuah titik pada bidang n-dimensi dinyatakan dalam bentuk vektor homogen  $(x, y, z, w)$ , di mana  $x$ ,  $y$ , dan  $z$  adalah koordinat euclidean, dan  $w$  adalah koordinat homogen. Sebuah matriks homografi  $H$  juga dinyatakan dalam bentuk matriks homogen:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = H \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (2-6)$$

### 2.2.7 Pembelajaran Mesin

Pembelajaran mesin adalah sebuah cabang dari kecerdasan buatan yang memungkinkan sistem komputer untuk belajar dari data, mengidentifikasi pola, dan melakukan prediksi dalam menyelesaikan permasalahan. Secara umum, pembelajaran mesin mencoba untuk menemukan pola tersembunyi dalam data dan mempergunakan informasi tersebut untuk menghasilkan hasil yang lebih baik dalam setiap iterasinya [22], [23], [24].

Sistem pembelajaran mesin dilatih dengan menggunakan data masukan dan berbagai teknik pemrosesan data untuk menghasilkan output yang diharapkan. Proses pembelajaran ini melibatkan pembuatan model matematis dan analisis data. Dalam pembelajaran mesin, data digunakan untuk melatih model atau algoritma, sehingga sistem dapat belajar dan meningkatkan kinerjanya dalam memecahkan masalah atau menyelesaikan tugas tertentu. Secara umum, ada tiga jenis pembelajaran mesin, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. *Supervised learning* melibatkan penggunaan data yang telah dilabeli atau dikategorikan sebelumnya untuk melatih model atau algoritma. *Unsupervised learning* melibatkan penggunaan data yang belum dilabeli atau dikategorikan sebelumnya untuk menemukan pola atau struktur yang terdapat dalam data. Sedangkan *reinforcement learning* melibatkan penggunaan sistem *reward* dan *punishment* untuk melatih model atau algoritma.

Dalam praktiknya, pembelajaran mesin sering digunakan dalam berbagai aplikasi bisnis, seperti analisis data, personalisasi produk, dan deteksi kecurangan. Dalam kehidupan sehari-hari, pembelajaran mesin juga diterapkan dalam pengenalan suara, identifikasi gambar, dan pengembangan sistem kendali otomatis.

## 2.3 Analisis Perbandingan Metode

## **BAB III**

### **METODE PENELITIAN**

Secara umum, tujuan dari penelitian ini adalah menguji apakah CDP dengan dua dan empat level *grayscale* dapat digunakan untuk autentikasi SQR orisinal atau palsu. Selain itu, penelitian ini juga menguji apakah penggunaan delapan ArUco *marker* di sekitar CDP dapat membantu dalam melokalisasi objek CDP. Penelitian ini juga mencari parameter yang digunakan dalam pembuatan SQR, seperti model SQR, konfigurasi kamera, serta jenis *printer* dan tinta yang digunakan untuk mencetak SQR. Dalam prosesnya, metode-metode penelitian yang dirancang dan dilakukan penulis akan dijelaskan pada bab ini.

#### **3.1 Alat dan Bahan Tugas Akhir**

##### **3.1.1 Alat Tugas Akhir**

Alat yang digunakan pada penelitian ini terbagi atas perangkat keras dan perangkat lunak yang dengan rincian sebagai berikut:

1. Laptop Lenovo Legion Y530

Merupakan perangkat keras utama yang digunakan dalam penelitian. Laptop ini memiliki spesifikasi sebagai berikut: Prosesor Intel I5-8300H, RAM 8 GB DDR4, dan kartu grafis Nvidia Geforce GTX 1050TI (4 GB).

2. Python 3.9

Perangkat lunak yang merupakan bahasa pemrograman utama yang digunakan dalam mengembangkan model SQR, pengolahan data, dan membuat model untuk mengautentikasi SQR.

3. AutoGluon

Merupakan pustaka AutoML sumber terbuka yang mengotomasi pembuatan model *deep learning* (DL) dan *machine learning* (ML) untuk menyelesaikan permasalahan di dunia nyata yang melibatkan *dataset* gambar, teks, dan tabular. AutoGluon dapat menyederhanakan alur kerja pembuatan model yang biasanya kita lakukan. Dengan AutoGluon, kita dapat mengembangkan dan memperbaiki model hanya dengan mengoptimasi parameter dengan beberapa baris kode saja. melakukan klasifikasi SQR asli dan palsu.

4. Microsoft Visual Studio Code

Merupakan salah satu perangkat lunak editor kode sumber terpopuler dan gratis yang dikembangkan oleh Microsoft. Editor ini memiliki berbagai fitur yang memungkinkan pengguna untuk menulis, mengedit, dan mengelola kode sumber de-

ngan mudah dan efisien. Selain itu, editor ini juga mendukung berbagai bahasa pemrograman, termasuk JavaScript, Python, Java, C++, dan masih banyak lagi. Tampilan antarmuka pengguna VS Code sangat bersih dan dapat disesuaikan, sehingga pengguna dapat mengatur tata letak dan tema yang sesuai dengan preferensi mereka. Selain itu, editor ini juga mendukung berbagai bahasa pemrograman, termasuk JavaScript, Python, Java, C++, dan masih banyak lagi. Fitur-fitur kunci dari VS Code termasuk kemampuan untuk mengeksekusi kode secara langsung dari editor, penyelesaian otomatis kode, refactoring kode, debugging, pengelolaan paket, dan integrasi dengan sistem kontrol versi seperti Git.

#### 5. Jupyter Notebook *Ekstension for VS Code*

Merupakan perangkat lunak ekstensi untuk visual studio code untuk menjalankan jupyter notebook di dalam *environment* visual studio code.

#### 6. Adobe Acrobat DC

Merupakan perangkat lunak yang digunakan sebagai pembaca dan pengedit fail PDF. Perangkat lunak ini banyak penulis gunakan untuk mengubah ukuran *batch QR* menjadi ukuran standar percetakan.

#### 7. Adobe Photoshop 2022

Merupakan perangkat lunak yang digunakan sebagai editor gambar. Perangkat lunak ini penulis gunakan dalam penelitian untuk mengubah fail PNG hasil *generate batch QR* dari kode menjadi PDF.

#### 8. *Smartphone* Realme GT Neo 3T

Perangkat keras yang digunakan sebagai pemindai SQR dalam pembuatan *dataset* SQR asli dan palsu. *Smartphone* ini memiliki spesifikasi sebagai berikut: *Chipset* Qualcomm SM8250-AC Snapdragon 870 5G, sistem operasi Android 12, kamera utama dengan resolusi 64 MP, f/1.8, 25mm.

#### 9. *Printer* Xerox Versant

Perangkat keras yang digunakan sebagai pencetak utama *dataset* SQR asli dan palsu. *Printer* ini merupakan *printer* berspesifikasi tinggi yang banyak digunakan dalam industri percetakan.

#### 10. Boks

Boks digunakan sebagai *environment* yang digunakan untuk menstandarisasi hasil pemotretan atau pemindaian *dataset* SQR menggunakan kamera *smartphone*.

#### 11. Gunting

Gunting digunakan untuk memotong *dataset* SQR yang nantinya akan dipotret.

### **3.1.2 Bahan Tugas akhir**

Bahan yang digunakan pada penelitian secara garis besar merupakan *dataset* SQR dengan rincian sebagai berikut:

- 1. Dataset SQR dengan Dua Level *Grayscale***

*Dataset* SQR dengan dua level *grayscale* terdiri dari 200 *dataset* foto pertama (original) dan 200 *dataset* foto kedua (palsu).

- 2. Dataset SQR dengan Empat Level *Grayscale***

*Dataset* SQR dengan empat level *grayscale* terdiri dari 200 *dataset* foto pertama (original) dan 200 *dataset* foto kedua (palsu).

- 3. Dataset SQR untuk Pengujian Parameter**

*Dataset* ini digunakan untuk mencari parameter pembuatan model, pemotretan, dan pencetakan SQR, misalnya ukuran penanda ArUco, ukuran SQR, ukuran CDP, konfigurasi kamera, dan jenis kertas dan tinta yang digunakan untuk mencetak SQR.

## **3.2 Tahapan Penelitian**

Penelitian ini diawali dengan melakukan studi literatur penelitian khususnya tentang SQR dan CDP. Selain itu, penulis juga banyak melakukan studi mengenai penanda ArUco untuk membantu lokalisasi objek, transformasi homografi, *template matching*, koefisien jarak, uji statistik T-Test untuk menguji signifikansi pada data grup, dan juga AutoML untuk membuat model klasifikasi biner SQR original atau palsu.

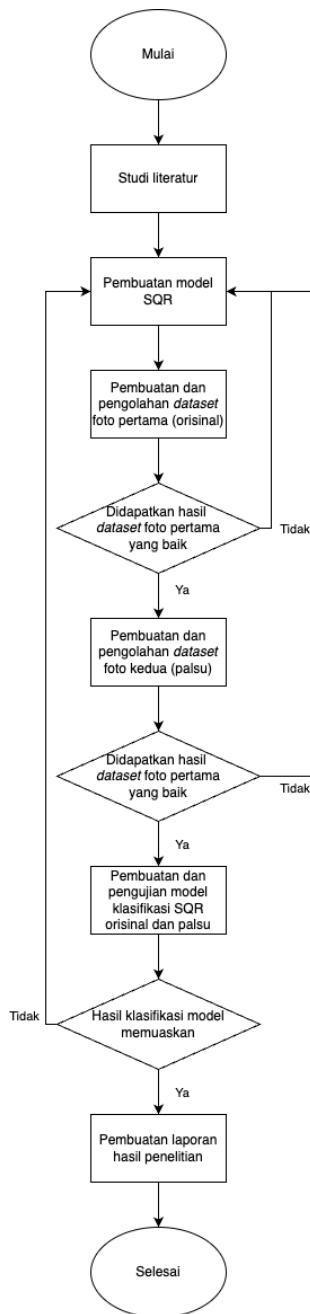
Selanjutnya, penulis melakukan desain dan pembuatan model SQR. Percobaan pembuatan desain SQR terdiri dari penentuan ukuran SQR dan toleransi kerusakannya, ukuran *watermark*, pembuatan CDP, dan juga penentuan jumlah, letak, jenis, dan ukuran penanda ArUco. Penentuan desain dan parameter-parameter tersebut dilakukan dengan cara studi literatur dari penelitian terdahulu dan juga pembuatan dan pengolahan dataset pengujian parameter.

Pembuatan dan pengolahan dataset foto pertama dan foto kedua dibagi menjadi beberapa tahapan. Beberapa tahapan tersebut digunakan untuk mencari parameter terbaik dalam pembuatan SQR. Parameter yang didapatkan adalah parameter konfigurasi kamera, jenis kertas dan tinta yang digunakan untuk mencetak SQR, ukuran penanda ArUco. Setelah parameter terbaik didapatkan, barulah penulis melakukan pembuatan *dataset* dengan jumlah data yang besar. *Dataset* yang digunakan untuk pembuatan model klasifikasi SQR original dan palsu adalah sebanyak 200 data original dan palsu 2 level serta 200 data original dan palsu 4 level, sehingga total data yang dibuat adalah sebanyak 800 data.

Setelah *dataset* dibuat, selanjutnya adalah pembuatan model klasifikasi SQR ori-

sinal dan palsu. Fitur yang digunakan dalam pembuatan model adalah *distance* atau jarak dari data orisinal atau data palsu dengan *template* yang telah di-*generate* pada proses pengolahan data sebelumnya. Pustaka pembelajaran mesin yang penulis gunakan dalam pemodelan klasifikasi biner adalah AutoGluon.

Setelah model selesai dibuat, selanjutnya adalah melakukan pengujian akurasi model dalam mengklasifikasikan SQR orisinal dan palsu. Setelah seluruh hasil penelitian dinilai memuaskan oleh penulis, tahapan terakhir adalah pembautan laporan hasil penelitian. Secara umum, alur penelitian yang dilakukan oleh penulis dapat dilihat pada Gambar 3.1



Gambar 3.1. Diagram alir tahapan penelitian

### **3.3 Pembuatan Model SQR**

#### **3.3.1 Penentuan Ukuran dan Versi Kode QR**

Dalam penelitian ini, versi kode QR yang digunakan oleh penulis adalah versi 3. Hal tersebut berdasarkan kebutuhan data yang akan digunakan dalam pembuatan dataset tidak lebih dari 35 karakter alfanumerik, sehingga kode QR versi 3 dengan ukuran modul 29x29 dapat memenuhi kebutuhan tersebut. Selain itu, adapun parameter *box-size* dan *border*, yaitu jumlah piksel dari tiap-tiap modul pada kode QR dan lapisan pembatas di luar kode QR. Pada pembuatan model SQR ini, penulis menetapkan ukuran *box-size* 20 piksel, dan ukuran *border* 3. Setelah itu penulis menambahkan *padding* tambahan berukuran 10 piksel di sisi terluar. Dengan demikian, ukuran total dari SQR adalah  $(29 \times 20) + (2 \times (3 \times 20)) + (2 \times 10) = 720 \times 720$  piksel.

#### **3.3.2 Penentuan Level Toleransi Kerusakan Kode QR**

Dalam pembuatan SQR, nantinya CDP akan ditempelkan atau diletakkan di tengah kode QR. Oleh karena itu, dibutuhkan level toleransi kerusakan yang besar. Diketahui bahwa level toleransi kerusakan tertinggi yang ada pada kode QR adalah level H dengan toleransi kerusakan kurang lebih 30%. Oleh karena itu, level toleransi kerusakan yang dipilih dalam model SQR ini adalah level H.

#### **3.3.3 Penentuan Ukuran Watermark**

Dengan level toleransi kode QR yang ditentukan adalah level H yang mana menerima kerusakan kurang lebih 30% kerusakan kode QR, maka ukuran piksel yang dapat dirusak adalah  $0,3 \times 29 = 8,7$ . Karena hasilnya tidak bulat, penulis melakukan pembulatan ke atas, sehingga ukuran piksel yang dapat dirusak adalah 9 piksel. Setelah dikalikan dengan *box-size*, maka ukuran toleransi kerusakan untuk meletakkan watermark adalah  $9 \times 20 = 180 \times 180$  piksel.

#### **3.3.4 Pembuatan CDP**

Setelah menentukan ukuran *watermark*, selanjutnya adalah mendesain CDP yang digunakan sebagai komponen autentikasi SQR. Dari hasil studi literatur penulis pada penelitian sebelumnya, CDP yang dinilai efektif adalah berukuran 100x100 piksel [1]. Semakin besar ukuran CDP, pola unit terkecilnya akan menjadi semakin besar. Hal ini menyebabkan informasi yang ada pada CDP tidak terdegradasi maksimal akibat proses P&S. Ukuran CDP 100x100 piksel dinilai optimal karena tetap sulit untuk dibuat ulang, namun ukuran unit terkecilnya masih kecil.

Penulis di sini akan menggunakan CDP dengan 2 dan 4 level yang nantinya akan dianalisis hasilnya. CDP di-*generate* berdasarkan DATA + SECRET. Hal ini merupakan

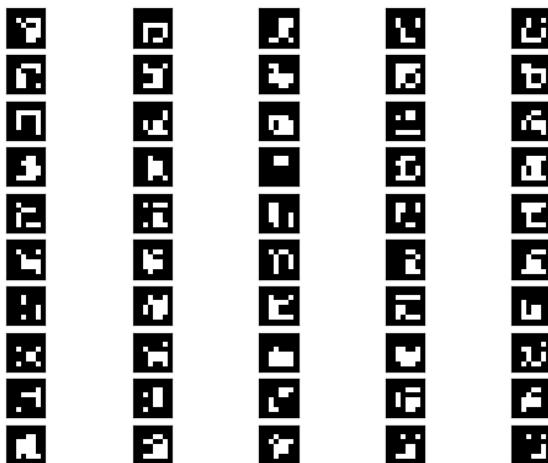
salah satu standar keamanan yang banyak digunakan saat ini, menggunakan kombinasi DATA + SECRET yang di-*hash* dan menghasilkan sebuah unique identifier. DATA merupakan nilai data yang tersimpan dalam kode QR, sedangkan SECRET merupakan string rahasia yang disimpan oleh pengembang SQR. DATA + SECRET akan di-*hash* menggunakan algoritma SHA1, kemudian keluarannya diset menjadi heksadesimal. Kemudian batasan nilainya diset menjadi UINT32 atau memiliki rentang nilai 0 s.d. 4294967296.

Setelah *seed* dibuat yang merupakan keluaran dari fungsi sebelumnya, selanjutnya adalah membuat CDP berukuran 100x100 piksel dengan 2 level atau 4 level *grayscale*. Untuk 2 level, nilai *grayscale*-nya adalah [0, 255] dan untuk 4 level, nilai *grayscale*-nya adalah [0, 85, 170, 255]. Nilai-nilai tersebut didefinisikan secara otomatis hanya dengan memasukkan nilai parameter *quant* saja. Kemudian, untuk memastikan tiap-tiap nilai memiliki distribusi yang seimbang dalam CDP, penulis membatasi perbandingan distribusi maksimal antar nilai adalah 1:3, misal untuk 4 level perbandingan distribusi yang dapat dipilih adalah [[1, 1, 1, 1], [3, 1, 2, 1], ..., [1, 1, 3, 1]]. Program nantinya akan memilih secara acak dari 10 variasi distribusi yang ada. Kemudian barulah CDP berukuran 100x100 piksel di-*generate* secara acak berdasarkan nilai *seed*, sehingga nilai *seed* akan berpengaruh pada distribusi elemen dan letak tiap-tiap elemen dalam CDP.

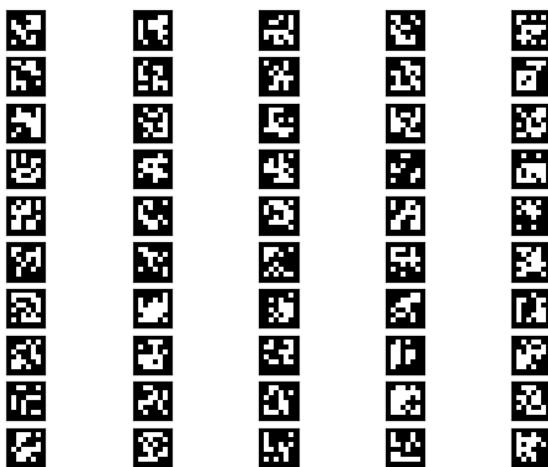
### 3.3.5 Menentukan Jumlah, Jenis, Letak, dan Ukuran Penanda ArUco

Setelah CDP dibuat, ruang kosong yang tersisa di dalam *watermark* adalah  $180 - 100 = 80$  piksel. Sisa ruang kosong tersebut digunakan untuk meletakkan penanda ArUco sebagai pembantu dalam melokalisasi objek CDP yang akan digunakan untuk autentikasi SQR. Jumlah penanda ArUco yang digunakan di sini sebanyak 8 buah. Kedelapan penanda ArUco diletakkan di area kosong di sekitar CDP, pojok kiri atas, tengah atas, pojok kanan atas, kiri tengah, kanan tengah, pojok kiri bawah, tengah bawah, dan pojok kanan bawah. Kedelapan penanda tersebut dinilai penulis cukup untuk merepresentasikan dan mewakili titik-titik yang nantinya akan digunakan untuk transformasi homografi. Penelitian sebelumnya menggunakan empat titik, yaitu keempat titik sudut kode QR. Hal tersebut memiliki kelemahan, yaitu jika ada *bending* pada foto pada bagian sisi-sisi kode QR (misal kode QR ditempelkan pada permukaan tabung), maka hasil transformasi homografinya akan kurang maksimal. Selanjutnya, untuk memudahkan program dalam mendeteksi kedelapan penanda ArUco, maka tentunya pola ArUco yang paling sederhana dapat memudahkan program dalam pendekripsi. Oleh karena itu, penulis memilih ArUco dengan ukuran modul 4 piksel. Pada Gambar 3.2 dan Gambar 3.3 dapat dilihat bahwa ArUco dengan ukuran modul 4 piksel memiliki pola yang relatif sederhana, sehingga toleransi degradasi gambar akibat proses P&S dapat diminimalisir.

Terakhir, adalah menentukan ukuran penanda ArUco yang diletakkan di sekitar CDP. Metode yang digunakan untuk mendapatkan ukuran adalah dengan membuat *da-*



Gambar 3.2. Sampel 50 penanda ArUco dengan ukuran modul 4x4 piksel



Gambar 3.3. Sampel 50 penanda ArUco dengan ukuran modul 6x6 piksel

taset sebanyak 40 data yang tiap 5 datanya memiliki ukuran ArUco 20 s.d. 34 piksel. Kemudian nantinya akan diuji, penanda ArUco dengan ukuran berapakah yang memiliki akurasi lokalisasi paling tinggi yang berarti kedekatan penanda ArUco dapat terdeteksi oleh program. Tiap-tiap penanda ArUco memiliki id unik yang membedakan penanda satu dengan lainnya dalam sebuah pustaka penanda ArUco. Oleh karena itu, penulis menggunakan id penanda ArUco tetap, yaitu 0 s.d. 7 untuk memudahkan pemetaan titik asal ke titik tujuan saat melakukan transformasi homografi, misalnya titik tujuan pojok kiri atas harus ditempati oleh penanda ArUco dengan id=0, titik pojok kanan bawah harus ditempati oleh marker dengan id=7 dan seterusnya.

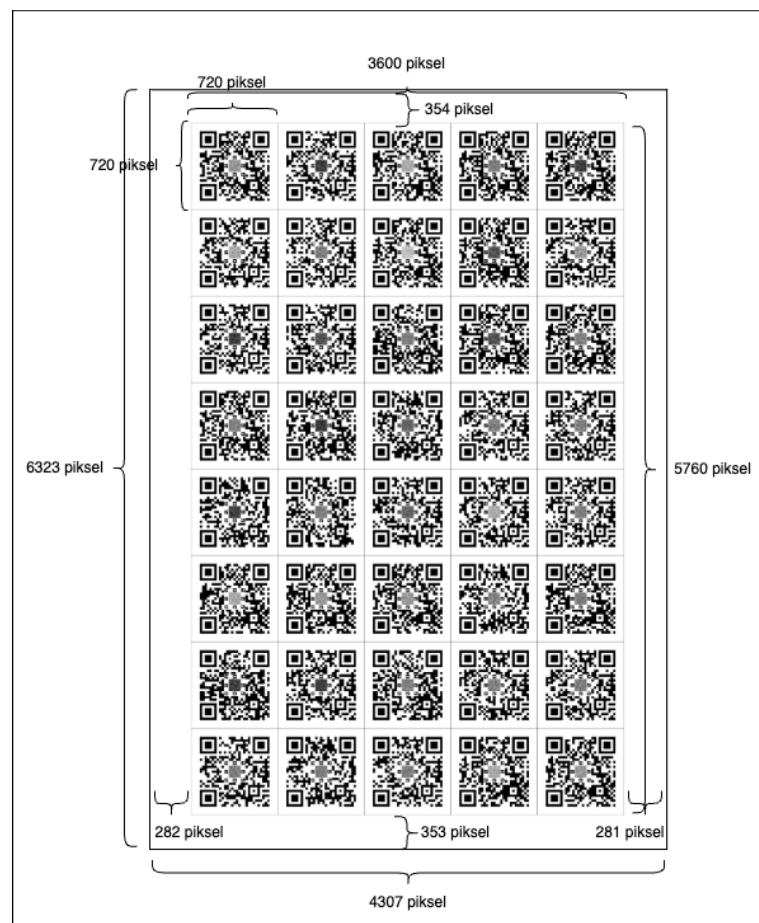
### 3.3.6 Menempelkan Watermark ke SQR

Setelah seluruh komponen *watermark* berukuran 180x180 piksel selesai dibuat, selanjutnya adalah menempelkannya atau meletakkannya ke tengah-tengah kode QR yang telah dibuat di awal.

### 3.4 Pembuatan dan Pengolahan Dataset SQR Foto Pertama (Orisinal)

#### 3.4.1 Pembuatan Batch SQR Foto Pertama (Orisinal)

SQR berjumlah 40 akan disusun menjadi 1 *batch*, dengan konfigurasi 8 baris 5 kolom. Diketahui juga bahwa ukuran satu SQR yang telah dirancang adalah 720x720 piksel. Nantinya SQR akan dirancang untuk dicetak dengan ukuran 55x55 mm, sehingga nantinya jumlah piksel per milimeter pada batch QR adalah  $720/55 = 13$  ppmm. Kemudian, untuk menyesuaikan ukuran piksel terhadap milimeter, perlu dilakukan penskalaan terhadap ppmm serta pemberian *padding*. Diketahui ukuran kertas A3+ adalah 329x483 mm, jika dijadikan ukuran piksel dengan 13 ppmm, maka ukurannya menjadi 4307x6323 piksel. Kemudian *padding* horizontal diset sebesar  $(6323-(720 \times 5))/2 = 281,5$  sedangkan *padding* vertikal diset sebesar  $(4307-(720 \times 8))/2 = 353,5$ . Karena hasilnya tidak bulat, maka *padding* horizontal dan vertikalnya diset ke +- bilangan bulat terdekat. Hasil akhir *batch* dalam ukuran piksel dapat dilihat pada Gambar 3.4.

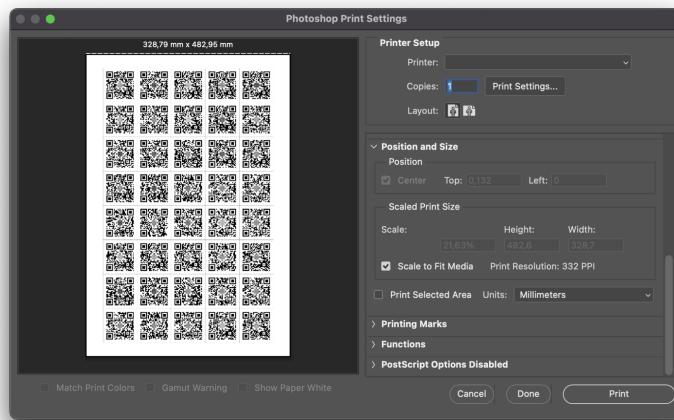


Gambar 3.4. *Batch* SQR orisinal (foto pertama) dalam piksel

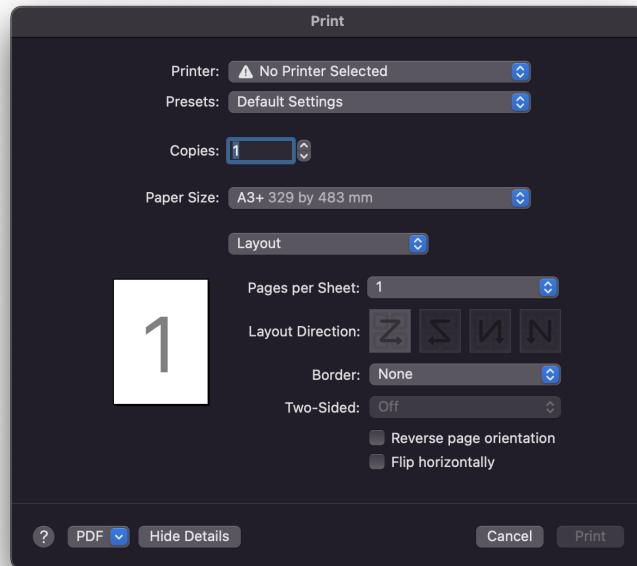
#### 3.4.2 Pencetakan Batch SQR Foto Pertama (Orisinal)

Dari hasil *generate batch* oleh program yang berukuran piksel dan berformat PNG, hasil tersebut masih perlu di-*scaling* menjadi ukuran kertas A3+ dengan format

PDF, sesuai dengan standar percetakan. Untuk melakukannya, penulis menggunakan perangkat lunak tambahan, yaitu Adobe Photoshop. Caranya adalah meng-*import batch* PNG ke dalam Photoshop, kemudian tekan CTRL + P untuk melakukan pencetakan. Kemudian di panel konfigurasi pencetakan, centang bagian *Scale to Fit Media*, hal ini bertujuan untuk melakukan penskalaan menuju ukuran kertas yang dituju. Selanjutnya, jika belum ada, masukkan konfigurasi untuk ukuran kertas A3+, yaitu dengan ukuran 329x483 mm. Terakhir adalah ubah target cetak menjadi PDF. Langkah-langkah lengkap konfigurasi pencetakan dan ukuran kertas dapat dilihat pada Gambar 3.5 dan Gambar 3.6.



Gambar 3.5. Konfigurasi saat melakukan pencetakan menggunakan Adobe Photoshop



Gambar 3.6. Konfigurasi menambahkan ukuran kertas A3+ dan cetak menjadi PDF

Terakhir, untuk jenis kertas dan tinta yang digunakan haruslah bersifat *doff*. Hal ini supaya hasil pemotretan tidak memantulkan cahaya ketika menggunakan flash dari kamera.

### 3.4.3 Pemotretan Batch SQR Foto Pertama (Orisinal)

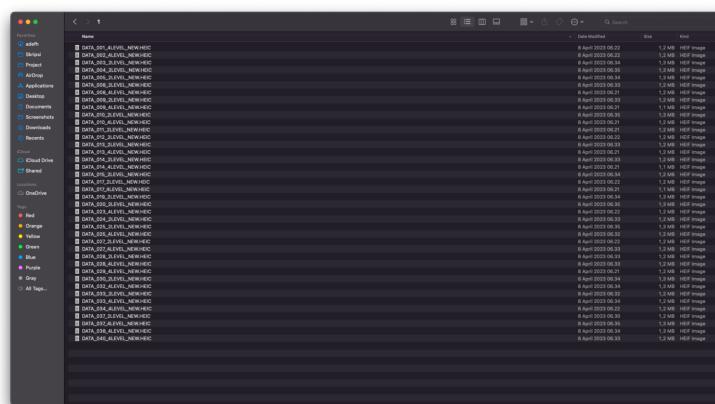
### 3.4.4 Lokalisasi CDP dari Raw Foto Pertama (Orisinal)

#### 3.4.4.1 Mengganti Nama Fail

Untuk melakukan lokalisasi CDP dari *dataset* foto, langkah pertama yang dilakukan adalah mengganti nama fail foto sesuai dengan data yang ada dalam kode QR. Hal ini bertujuan untuk memudahkan pengelolaan *dataset*. Nama fail sebelum dan sesudah perubahan nama dapat dilihat pada Gambar 3.7 dan Gambar 3.8.



Gambar 3.7. Nama fail sebelum perubahan



Gambar 3.8. Nama fail sebelum perubahan

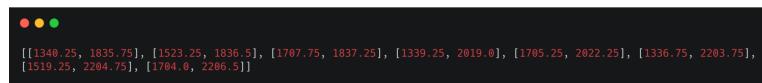
Untuk mengganti nama fail, penulis memanfaatkan pustaka OpenCV yang digunakan untuk mendeteksi data dalam kode QR. Pustaka tersebut juga dapat digunakan untuk mendeteksi koordinat keempat titik sudut kode QR yang digunakan dalam penelitian sebelumnya untuk melakukan transformasi homografi dari titik asal ke titik tujuan. Sebelum dideteksi dan di-*decode* gambar foto *raw* akan diproses terlebih dahulu, untuk meningkatkan performa deteksi oleh pustaka QRDecoder. Pemrosesan yang dilakukan yaitu, mengubah gambar menjadi *grayscale* dan menambahkan filter GaussianBlur pada gambar yang telah diubah menjadi *grayscale* dengan ukuran kernel tertentu. Hal

ini bertujuan untuk menghilangkan derau dari gambar. Setelah itu, gambar akan dilakukan perulangan scaling dengan faktor skala 1 s.d. 0,1. Pada tiap-tiap faktor skala, fungsi QRDecoder akan mencoba mendeteksi keempat titik sudut kode QR dan data di dalamnya. Jika titik sudut yang dideteksi sudah berjumlah 4 dan data di dalamnya sudah terdeteksi, maka proses penskalaan akan dihentikan pada faktor skala tertinggi, lalu kemudian data yang tersimpan pada kode QR akan dikeluarkan. Misal, saat perulangan dengan faktor skala 1, empat titik sudut pada kode QR sudah terdeteksi dan datanya juga, jika hasilnya demikian, artinya gambar tidak perlu di-*scaling* untuk mendeteksi dan men-*decode* kode QR. Namun, apabila perulangan berhenti di faktor skala 0,5, artinya gambar perlu di-*scaling* setengahnya untuk dapat dideteksi dan di-*decode* datanya.

#### 3.4.4.2 Lokalisasi CDP dari Foto

Untuk melakukan lokalisasi CDP dari foto yang telah diganti nama fail-nya, penulis memanfaatkan penanda ArUco yang diletakkan di sekitar CDP. Telah dijelaskan sebelumnya, bahwa delapan penanda ArUco dapat melakukan lokalisasi CDP dengan lebih baik karena titik yang digunakan untuk transformasi homografi dari titik asal ke titik tujuan lebih banyak dari empat titik yang didapatkan dari fungsi QRDecoder. Hal ini dapat mengatasi masalah *bending* gambar pada kondisi tertentu. Sama seperti pemrosesan sebelumnya, gambar akan diubah menjadi *grayscale* dan ditambahkan filter GaussianBlur untuk meningkatkan performa deteksi penanda ArUco. Kemudian, gambar akan dilakukan perulangan penskalaan dengan faktor skala 1 s.d. 0,1. Pada tiap-tiap faktor skala, fungsi aruco.detectMarkers akan mendeteksi titik-titik sudut dari kedelapan penanda ArUco beserta id-nya. Sama seperti pendekripsi data pada kode QR, perulangan akan dihentikan apabila delapan penanda ArUco sudah berhasil dideteksi.

Dari keluaran empat titik sudut di tiap-tiap penanda, penulis memprosesnya untuk menjadikannya satu titik di tengah untuk memudahkan pendefinisian titik asal transformasi. Kemudian hasilnya akan diolah menjadi *xy\_list*, yaitu koordinat titik tengah dari masing-masing penanda mulai dari id=0 s.d. id=7. Contoh *xy\_list* yang telah diolah dapat dilihat pada Gambar 3.9.



Gambar 3.9. *xy\_list* yang menyimpan koordinat titik tengah penanda ArUco dari id 0 s.d. 7

Untuk contoh plot hasil deteksi koordinat ke dalam *xy\_list* pada gambar hasil foto dapat dilihat pada Gambar 3.10.

Setelah semua titik telah didapatkan, selanjutnya adalah melakukan transformasi homografi dari kedelapan titik tersebut ke titik tujuan. Untuk menentukan titik tujuan,



Gambar 3.10. Hasil plot koordinat titik tengah penanda ArUco pada xy\_list ke gambar hasil foto

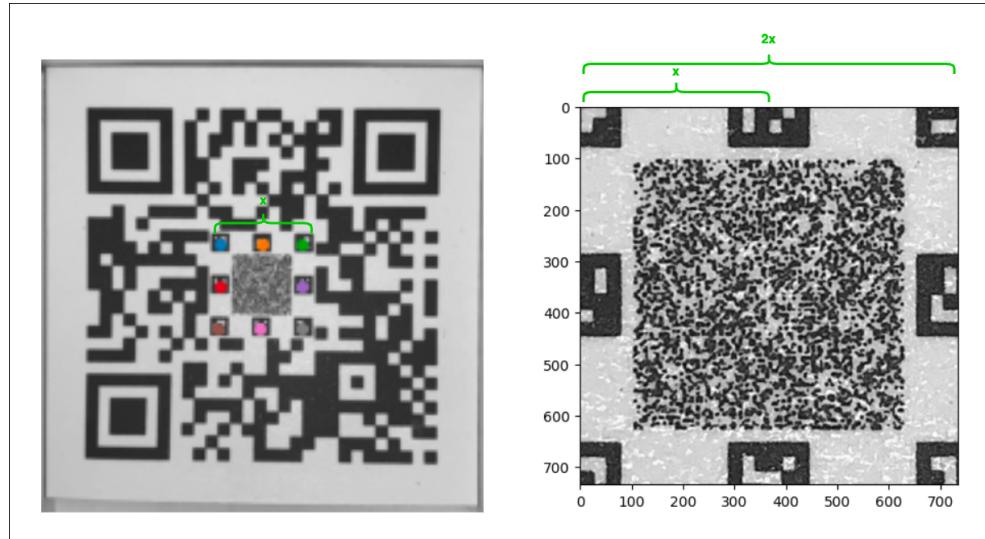
penulis menggunakan acuan ukuran dari jarak terdekat dari empat titik penanda ArUco yang terletak di pojok (pojok kiri atas, pojok kanan atas, pojok kiri bawah, dan pojok kanan bawah) atau (id 0, id 2, id 5, dan id 7) menggunakan *euclidian distance*. Hal ini digunakan untuk menjaga kualitas hasil dari transformasi homografi. Dari eksperimen, jika ukuran tujuan transformasi homografi terlalu jauh dari ukuran awal, misalnya langsung didefinisikan berukuran 100x100 piksel sesuai *template* hasilnya akan rusak. Setelah jarak terdekatnya didapatkan sebagai x, selanjutnya adalah menentukan titik tujuan. Titik tujuan transformasi didefinisikan dalam bentuk sebuah *list* seperti yang terlihat pada Gambar 3.11, dimana x merupakan jarak terdekat dari penanda ArUco yang terletak di pojok.



Gambar 3.11. List koordinat tujuan transformasi homografi

Dalam melakukan transformasi homografi, koordinat tujuan nilainya dikalikan dengan 2, sehingga ukuran dari CDP terlokalisasi adalah 2 kali dari CDP awal sebelum dilokalisasi. Hal ini bertujuan untuk menjaga kualitas CDP yang nantinya masih akan

diproses lagi. Untuk melihat perbedaan SQR awal dan SQR setelah ditransformasi homografi berdasarkan koordinat dari marker dapat dilihat pada Gambar 3.12.



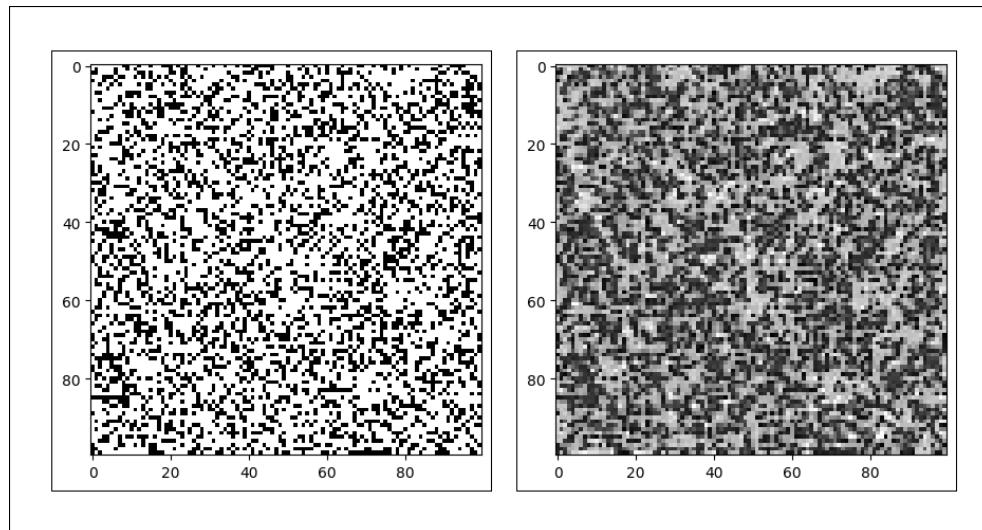
Gambar 3.12. SQR awal dan setelah dilakukan transformasi homografi

Setelah melakukan transformasi homografi, langkah selanjutnya adalah mendapatkan objek CDP itu sendiri. Diketahui bahwa desain pada desain SQR awal, CDP berukuran 100x100 piksel, sedangkan ukuran gambar watermark hingga titik tengah penanda ArUco adalah 140x140 piksel. Oleh karena itu, untuk mendapatkan objek CDP dari objek hasil transformasi adalah dengan melakukan pemotongan di tengah sebesar 100/140 bagian dari objek hasil transformasi. Setelah objek CDP didapatkan, langkah terakhir adalah melakukan penskalaan CDP menjadi 100x100 piksel sesuai dengan ukuran *template*. CDP hasil lokalisasi berukuran 100x100 piksel inilah yang akan diolah datanya untuk dibandingkan dengan *template*. Perbandingan CDP *template* dengan CDP hasil akhir dari lokalisasi dapat dilihat pada Gambar 3.13. Kemudian untuk memudahkan manajemen fail CDP sebagai *dataset*, hasil lokalisasi disimpan menjadi sebuah fail gambar berformat PNG dengan format penamaan fail CDP\_{data}.png. Hasil akhir dataset CDP foto pertama (orisinal) dikumpulkan ke dalam sebuah folder seperti pada Gambar 3.14.

### 3.4.5 Pembuatan Fitur Jarak (Orisinal dengan *Template*)

Untuk mendapatkan fitur dari *dataset* foto pertama (orisinal) yang nantinya digunakan dalam pembuatan model klasifikasi menggunakan AutoML, penulis menggunakan jarak spasial, histogram, dan dct dengan beberapa koefisien jarak dari CDP foto pertama (orisinal) dengan *template*.

Langkah pertama adalah membaca seluruh fail CDP dalam folder yang sudah ditentukan, kemudian dilakukan *parsing* pengambilan jenis interpolasi dan data sesuai dengan format *dataset* CDP. Selanjutnya adalah men-generate *template* berdasarkan data



Gambar 3.13. Perbandingan CDP *template* dengan CDP hasil lokalisasi foto pertama (orisinal)

yang sudah di-*parsing*, simpan sebagai *template*. Setelah itu, baca CDP hasil scan, ubah menjadi *grayscale*, simpan sebagai *scanned*. Kemudian, tiap fail CDP *template* dan *scanned* akan diperbesar 4 kali pembesaran, dari awalnya 100x100 piksel di-*scaling* menjadi 400x400 piksel. Selanjutnya, matriks tersebut akan diolah menjadi tiga fitur, yaitu DCT, histogram, dan spasial. Kemudian akan dihitung jaraknya menggunakan koefisien jarak korelasi, kosinus, *euclidean*, dan *canberra* untuk tiap-tiap fitur. Untuk tiap perulangan (tiap pembacaan fail) hasil fitur jarak, data, jenis interpolasi, dan level (2 atau 4 level) akan di-*append* ke dalam sebuah *list*. Setelah perulangan selesai, *list* tersebut akan diubah menjadi sebuah *dataframe*, di-*sorting* berdasarkan jenis interpolasi dan datanya, lalu ditambahkan kolom label (orisinal atau palsu). Setelah *datagramme* siap, selanjutnya adalah mengekspor *datagramme* tersebut menjadi sebuah fail csv. Untuk hasil akhir *dataframe* sebelum diekspor menjadi fail csv dapat dilihat pada Gambar 3.15.

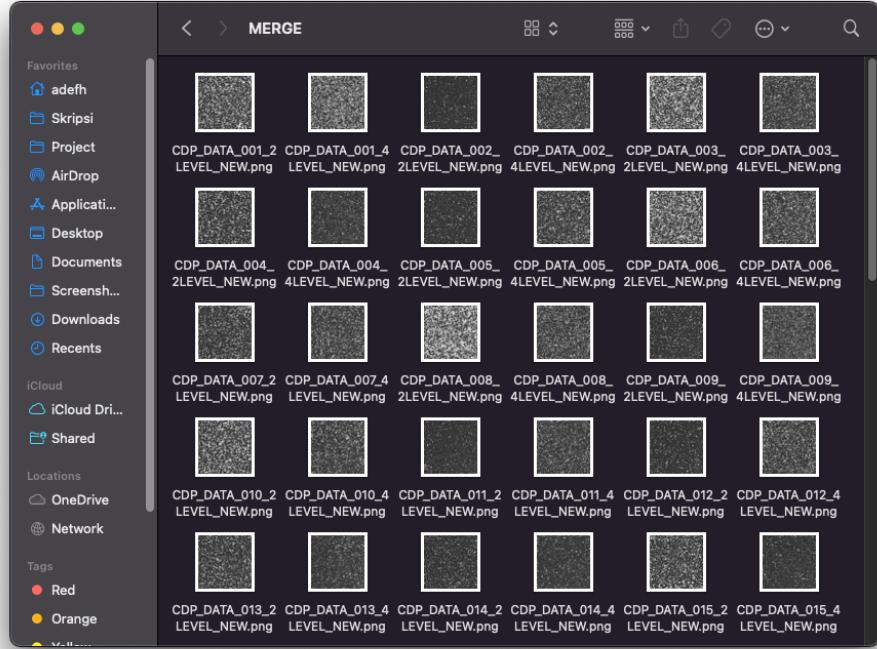
### 3.5 Pembuatan dan Pengolahan *Dataset SQR Foto Kedua (Palsu)*

#### 3.5.1 Pembuatan Model SQR Foto Kedua (Palsu)

Model SQR palsu dibuat dengan men-*generate template* sesuai data pada SQR orisinal. Kemudian CDP hasil lokalisasi foto pertama akan ditempelkan di tengah-tengah *template* SQR, sehingga perbedaan antara SQR orisinal dan palsu hanya ada pada CDP-nya saja. Hal ini tentu akan sulit dibedakan oleh mata manusia.

#### 3.5.2 Pembuatan *Batch SQR Foto Kedua (Palsu)*

Untuk pembuatan *batch* SQR palsu, langkahnya sama dengan pembuatan *batch* SQR foto pertama pada 3.3.1, hanya saja SQR yang digunakan adalah SQR yang telah diganti CDP-nya. Jumlah SQR dalam satu *batch* juga sama, 40 buah, dengan konfigurasi



Gambar 3.14. Hasil *dataset* CDP foto pertama (orisinal) yang sudah dilokalisasi dan diganti nama failnya

	interpolation	data	sp_corr	sp_cosine	sp_euclidean	sp_cantberra	his_corr	his_cosine	his_euclidean	his_cantberra	dct_corr	dct_cosine	dct_euclidean	dct_cantberra	level	label
0	INTER_AREA	DATA_001_2LEVEL_NEW	0.276514	0.105260	43818.564468	97199.766805	1.088972	1.000000	0.725821	211.000000	0.085241	0.065129	1.373189e+06	98948.011457	2LEVEL	ORI
1	INTER_AREA	DATA_001_4LEVEL_NEW	0.347229	0.065895	32862.214524	51246.687912	1.039011	0.932388	0.533133	221.916512	0.037199	0.037109	1.096547e+06	9.2848528126	4LEVEL	ORI
2	INTER_AREA	DATA_002_2LEVEL_NEW	0.423337	0.376830	40254.680145	141408.297314	1.030277	1.000000	0.810349	146.920000	0.289255	0.288948	1.144804e+06	124273.9386159	2LEVEL	ORI
3	INTER_AREA	DATA_002_4LEVEL_NEW	0.311012	0.122413	36713.780781	77762.781353	1.037287	0.957346	0.521934	193.925391	0.075712	0.075599	1.122782e+06	102565.295199	4LEVEL	ORI
4	INTER_AREA	DATA_003_2LEVEL_NEW	0.251024	0.075142	41157.396225	83716.713034	1.122129	1.000000	0.752067	226.000000	0.044829	0.044738	1.328214e+06	900079.84576	2LEVEL	ORI
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	INTER_NEAREST	DATA_038_2LEVEL_NEW	0.848117	0.247521	35628.271246	85653.094117	1.029187	0.961729	0.539581	192.951196	0.149387	0.149172	1.077218e+06	119270.095176	4LEVEL	ORI
396	INTER_NEAREST	DATA_039_2LEVEL_NEW	0.879838	0.249665	56019.305387	95901.322562	1.136961	0.997535	0.749396	255.998402	0.147533	0.147237	1.708828e+06	115732.9388582	2LEVEL	ORI
397	INTER_NEAREST	DATA_039_4LEVEL_NEW	0.826814	0.182059	36858.80632	64068.041681	1.033648	0.938115	0.553876	240.946065	0.104194	0.104041	1.152029e+06	112314.488712	4LEVEL	ORI
398	INTER_NEAREST	DATA_040_2LEVEL_NEW	0.729232	0.244817	53257.576404	103062.861930	1.127558	0.997764	0.726175	254.999339	0.149274	0.148983	1.616388e+06	110465.427157	2LEVEL	ORI
399	INTER_NEAREST	DATA_040_4LEVEL_NEW	0.691139	0.209644	43803.895900	80447.623041	1.076971	0.957107	0.534011	250.896670	0.125290	0.125047	1.335360e+06	109099.753744	4LEVEL	ORI

Gambar 3.15. Dataframe fitur jarak *dataset* foto pertama (orisinal)

8 baris 5 kolom. Ukuran kertas juga disesuaikan menjadi A3+ sesuai standar percetakan.

### 3.5.3 Pencetakan *Batch SQR* Foto Kedua (Palsu)

Proses persiapan pencetakan juga sama dengan persiapan pencetakan *batch SQR* foto pertama (orisinal) pada 3.3.2, hasil *generate* yang ukurannya masih piksel, diubah menjadi mm dengan ukuran kertas A3+ menggunakan perangkat lunak Adobe Photoshop. Kertas yang digunakan untuk mencetak juga sama, yaitu yang bersifat *doff* dan tidak memantulkan cahaya. *Printer* yang digunakan untuk mencetak haruslah sama, hal ini untuk memastikan tidak adanya perbedaan kualitas akibat perbedaan perangkat *printer*.

### **3.5.4 Pemotretan *Batch SQR* Foto Kedua (Palsu)**

### **3.5.5 Lokalisasi CDP dari *Raw Foto* Kedua (Palsu)**

Proses lokalisasi CDP foto kedua (palsu) sama persis dengan proses lokalisasi CDP foto pertama (orisinal) pada 3.3.4. Perbedaanya adalah penamaan fail keluaran, yaitu dengan format CDP\_{interpolasi}\_{data}\_FAKE.

### **3.5.6 Pembuatan Fitur Jarak (Palsu dengan Templat)**

Proses ini juga sama seperti yang dilakukan pada pembuatan fitur jarak orisinal dengan *template*, hanya saja yang diukur jaraknya adalah CDP foto kedua (palsu) dengan *template*. Selain itu, kolom label pada *dataframe* nilainya adalah FAKE.

## **3.6 Pembuatan dan Pengujian Model Klasifikasi SQR Orisinal dan Palsu**

Model yang dibangun merupakan model klasifikasi biner, untuk mengklasifikasikan apakah SQR tersebut orisinal atau palsu. Pembuatan model menggunakan pustaka AutoML dari AutoGluon. AutoGluon dipilih karena dapat melakukan pemrosesan fitur otomatis dan memberikan peringkat *leaderboard* untuk model-model yang digunakan, sehingga penulis dapat dengan mudah memilih model dengan akurasi klasifikasi biner tertinggi.

### **3.6.1 Pemodelan Klasifikasi Biner menggunakan AutoGluon**

Penulis menggunakan pustaka AutoML dari AutoGluon dalam pembuatan model klasifikasi biner SQR orisinal dan palsu. Sebelum memasukkan data ke model, perlu dilakukan pemisahan antara X dan y, X merupakan kolom fitur sedangkan y merupakan kolom label. Kolom fitur yang dipilih adalah fitur jarak yang berjumlah 12 kolom. Kolom label adalah kolom bernilai ORI atau FAKE. Setelah dilakukan pemisahan X dan y, selanjutnya adalah memisahkan X\_train, X\_test, y\_train, dan y\_test. Perbandingan data latih dan data uji adalah 7:3. Setelah itu, data yang sudah dipisah dimasukkan ke dalam model AutoGluon. AutoGluon dapat secara otomatis menampilkan *leaderboard* untuk model-model yang digunakan.

## **3.7 Metode yang Digunakan**

### **3.7.1 Pembuatan Model SQR Orisinal**

Model SQR yang coba dikembangkan oleh penulis adalah menempelkan *watermark* di tengah-tengah kode QR biasa. *Watermark* yang ditempelkan merupakan gabungan dari CDP dan juga enam ArUco *markers* untuk membantu lokalisasi objek CDP. CDP yang ditempelkan pada kode QR digunakan untuk autentikasi, apakah SQR tersebut orisinal ataukah palsu. Secara umum, desain dari SQR yang dirancang oleh penulis dapat dilihat pada

### 3.7.1.1 Penentuan Versi Kode QR

Setiap versi dari kode QR memiliki kapasitas penyimpanan data yang berbeda, kode QR versi 1 memiliki ukuran  $21x21$  modul, sedangkan kode QR versi 40 memiliki ukuran  $177x177$  modul. Setiap kenaikan satu versi kode QR, maka akan ada penambahan sebanyak 4 modul, misalkan kode QR versi 1 memiliki jumlah modul  $21x21$ , maka kode QR versi 2 memiliki jumlah modul  $25x25$ , kode QR versi 3 memiliki jumlah modul  $29x29$ , dan seterusnya. Semakin banyak modul yang dimiliki kode QR, semakin banyak juga data yang dapat disimpan. Karena data yang digunakan dalam pengujian yang dilakukan adalah data berformat  $DATA_{160_x} LEVEL_{ARC30}$ , di mana  $x$  adalah kuantisasi level (2 dan 4), maka kode QR versi 3 dengan kapasitas maksimal menyimpan 35 karakter alfanumerik saat menggunakan level koreksi kesalahan H sudah cukup untuk menyimpan data tes.

### 3.7.1.2 Penentuan Level Koreksi Kesalahan

PCT level merupakan ukuran untuk seberapa banyak kode QR akan dirusak untuk ditempeli *watermark*. Karena *watermark* yang didesain berukuran  $172x172$  piksel, sedangkan ukuran kode QR dengan *box\_size* 20 dan jumlah modul 29 sesuai dengan jumlah modul kode QR versi 3. Karena *watermark* yang didesain berukuran  $172x172$ , maka level koreksi kesalahan yang digunakan adalah level H atau sama dengan 30% koreksi kesalahan yang ditolerir. Jumlah modul yang dapat dirusak atau ditolerir kerusakannya oleh koreksi kesalahan level H adalah:

$$\begin{aligned} n &= n_v \cdot b_s && (3-1) \\ &= 29 \cdot 20 \\ &= 580 \end{aligned}$$

dimana :

$$n = \text{Jumlah ukuran modul kode QR} \quad (3-3)$$

$$n_v = \text{Jumlah modul pada versi kode QR} \quad (3-4)$$

$$b_s = \text{Ukuran box size} \quad (3-5)$$

di mana:

$$n = \text{Ukuran piksel kode QR}$$

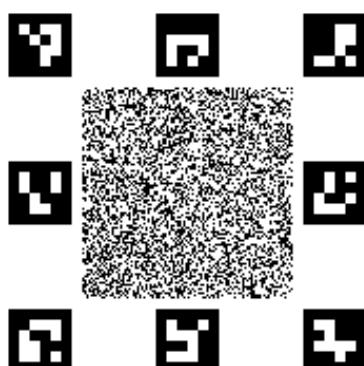
$$n_v = \text{Ukuran piksel kode QR}$$

### 3.7.1.3 Pembuatan CDP

CDP yang akan dibuat di sini menggunakan ukuran  $100 \times 100$  piksel, mengacu pada ukuran yang digunakan peneliti sebelumnya [1]. Dengan ukuran  $100 \times 100$  diharapkan degradasi informasi yang dihasilkan lebih besar dari proses P&S, sehingga model pembelajaran mesin dapat melakukan klasifikasi dengan baik. CDP di-*generate* secara random menggunakan sebuah *seed* yang nilainya adalah gabungan dari data dengan secret. *Seed* ini digunakan untuk memastikan CDP dapat di-*generate* dari perangkat yang berbeda, baik untuk melakukan pencetakan ataupun verifikasi. Secara umum, alur pembuatan CDP adalah sebagai berikut:

### 3.7.1.4 Pembuatan ArUco Marker di Sekitar CDP

Setelah CDP berukuran  $100 \times 100$  piksel di-*generate*, selanjutnya adalah meletakkan *marker* di sekitarnya. Caranya adalah dengan memberikan *padding* atau ruang kosong di sekitar CDP berukuran 40 piksel. Setelah itu, keenam *marker* di letakkan di tengah-tengah ruang kosong tersebut. *Marker* dengan id paling kecil diletakkan di pojok kiri atas, kemudian selanjutnya di letakkan di kanannya, hingga *marker* dengan id terakhir diletakkan di pojok kiri bawah. ArUco *marker* yang digunakan sendiri berukuran  $30 \times 30$  piksel. Ukuran tersebut didapatkan dari hasil percobaan oleh penulis, menguji beberapa ukuran *marker* dari  $20 \times 20$  piksel sampai  $30 \times 30$  piksel. Hal ini digunakan sebagai acuan lokasi untuk transformasi homografi, seadainya nanti SQR yang difoto posisinya diputar-putar. Gambaran akhir CDP yang telah dilengkapi enam ArUco *marker* dapat dilihat pada 3.16



Gambar 3.16. SQR diharapkan dapat mendeteksi kode QR palsu pada saat diautentikasi oleh pengguna [2]

- 3.7.1.5 Peletakan *Watetemark* di Kode QR**
- 3.7.1.6 Pembuatan *Batch SQR* Orisinal**
- 3.7.1.7 Pencetakan dan Pemotretan *Dataset SQR* Orisinal**
- 3.7.2 Pengujian Ukuran ArUco *Marker***
- 3.7.3 Mengukur *Distance* CDP1 dengan CDPT**
- 3.7.4 Uji Statistik Hasil Pengukuran *Distance***
- 3.7.5 Pembuatan Model SQR Palsu**
  - 3.7.5.1 Mengganti Nama dari Hasil Foto SQR Berdasarkan dengan Data SQR**
  - 3.7.6 Membuat Templat SQR Palsu Berdasarkan Data Nama Foto**
    - 3.7.6.1 Melokalisasi CDP Foto Pertama dan Menempelkannya ke SQR Palsu**
    - 3.7.6.2 Menyimpan SQR Palsu ke Dalam Sebuah Fail**
    - 3.7.6.3 Membuat *Batch SQR* Palsu**
    - 3.7.6.4 Model Klasifikasi SQR Orisinal dan Palsu**

### **3.8 Alur Tugas Akhir**

Menguraikan prosedur yang akan digunakan dan jadwal atau alur penyelesaian setiap tahap. Alur penelian ini dapat disajikan dalam bentuk diagram. Diagram dapat disusun dengan aturan yang baik semisal menggunakan *flowchart*. Aturan dan tutorial pembuatan *flowchart* dapat dilihat di <http://ugm.id/flowcharttutorial>. Setelah menggambarkannya, penulis wajib menjelaskan langkah-langkah setiap alur tugas akhir dalam sub bab tersendiri sesuai dengan kebutuhan.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Pembahasan Hasil 1 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)**

Poin pertama adalah membahas tujuan penelitian pertama. Dapat ditambahkan beberapa sub bab jika diperlukan.

#### **4.2 Pembahasan Hasil 2 (Ubah Judul Sesuai dengan Hal yang Hendak dibahas)**

Poin kedua adalah membahas tujuan penelitian kedua. Dapat ditambahkan beberapa sub bab jika diperlukan. Dapat juga diteruskan ke Sub Bab Pembahasan hasil 3 dan seterusnya, jika ada tiga atau lebih tujuan penelitian.

#### **4.3 Perbandingan Hasil Penelitian dengan Hasil Terdahulu**

Pembahasan penutup dapat menjelaskan mengenai kelebihan hasil pengembangan / penelitian dan kekurangan dibandingkan dengan skripsi atau penelitian terdahulu atau perbandingan terhadap produk lain yang ada di pasaran. Penulis dapat menggunakan tabel untuk membandingkan secara gamblang dan menjelaskannya.

## **BAB V**

### **TAMBAHAN (OPSIONAL)**

Anda boleh menambahkan Bab jika diperlukan. Jumlah Bab tidak harus sesuai dengan *template*.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Kesimpulan dapat diawali dengan apa yang dilakukan dengan tugas akhir ini lalu dilanjutkan dengan poin-poin yang menjawab tujuan penelitian, apakah tujuan sudah tercapai atau belum, tentunya berdasarkan data ataupun hasil dari Bab pembahasan sebelumnya. Dalam beberapa hal, kesimpulan dapat juga berisi tentang temuan/*findings* yang Anda dapatkan setelah melakukan pengamatan dan atau analisis terhadap hasil penelitian.

#### **6.2 Saran**

Saran berisi hal-hal yang bisa dilanjutkan dari penelitian atau skripsi ini, yang belum dilakukan karena batasan permasalahan. Saran bukan berisi saran kepada sistem atau pengguna, tetapi saran diberikan kepada aspek penelitian yang dapat dikembangkan dan ditambahkan di penelitian atau skripsi selanjutnya.

## DAFTAR PUSTAKA

- [1] E. Khermaza, I. Tkachenko, and J. Picard, “Can copy detection patterns be copied? evaluating the performance of attacks and highlighting the role of the detector,” in *2021 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2021, pp. 1–6.
- [2] J. Picard, P. Landry, and M. Bolay, “Counterfeit detection with qr codes,” in *Proceedings of the 21st ACM Symposium on Document Engineering*, 2021, pp. 1–4.
- [3] BASCAP and INTA, *The Economic Impacts of Counterfeiting and Piracy*. Business Action to Stop Counterfeiting and Piracy (BASCAP), 2016.
- [4] C. W. Hill, “Digital piracy: Causes, consequences, and strategic responses,” *Asia Pacific Journal of Management*, vol. 24, pp. 9–25, 2007.
- [5] B. Depoorter, “Intellectual property infringements & 3d printing: Decentralized piracy,” *Hastings LJ*, vol. 65, p. 1483, 2013.
- [6] R. L. Van Renesse, *Optical document security*. Artech House Publishers, 1998.
- [7] C. W. Wong and M. Wu, “Counterfeit detection based on unclonable feature of paper using mobile camera,” *IEEE Transactions on Information Forensics and Security*, vol. 12, 2017.
- [8] R. Schraml, L. Debiasi, and A. Uhl, “Real or fake: Mobile device drug packaging authentication,” in *Proceedings of the 6th ACM workshop on information hiding and multimedia security*, 2018, pp. 121–126.
- [9] J. Picard, “Digital authentication with copy-detection patterns,” in *Optical Security and Counterfeit Deterrence Techniques V*, vol. 5310. SPIE, 2004, pp. 176–183.
- [10] A. T. Phan Ho, B. A. Mai Hoang, W. Sawaya, and P. Bas, “Document authentication using graphical codes: Reliable performance analysis and channel optimization,” *EURASIP Journal on Information Security*, vol. 2014, pp. 1–17, 2014.
- [11] J. Picard, “On the security of copy detectable images,” in *NIP & Digital Fabrication Conference*, vol. 2008, no. 2. Society for Imaging Science and Technology, 2008, pp. 796–798.
- [12] A. E. Dirik and B. Haas, “Copy detection pattern-based document protection for variable media,” *IET Image Processing*, vol. 6, no. 8, pp. 1102–1113, 2012.
- [13] D. ADC, “Qr code essentials,” 2011.
- [14] J. Picard, C. Vielhauer, and N. Thorwirth, “Towards fraud-proof id documents using multiple data hiding technologies and biometrics,” in *Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306. SPIE, 2004, pp. 416–427.
- [15] C. Harwood, “Optical document security,” *Kybernetes*, vol. 27, no. 5, pp. 586–588, 1998.

- [16] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *Computer Vision, IEEE International Conference on*, vol. 3. IEEE Computer Society, 2003, pp. 1470–1470.
- [17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
- [18] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 39.
- [19] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.
- [20] S. Al-Anazi, H. Almahmoud, and I. Al-Turaiki, “Finding similar documents using different clustering techniques,” vol. 82, 2016.
- [21] P. H. Sneath, R. R. Sokal *et al.*, *Numerical taxonomy. The principles and practice of numerical classification.*, 1973.
- [22] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [23] E. Alpaydin, “Introduction to machine learning ethem alpaydin.” *Introduction to Machine Learning, Third Edition*, 2014.
- [24] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

**Catatan:** Daftar pustaka adalah apa yang dirujuk atau disitasi, bukan apa yang telah dibaca, jika tidak ada dalam sitasi maka tidak perlu dituliskan dalam daftar pustaka.

## **LAMPIRAN**

### **L.1 Isi Lampiran**

Lampiran bersifat opsional bergantung hasil kesepakatan dengan pembimbing dapat berupa:

1. Bukti pelaksanaan Kuesioner seperti pertanyaan kuesioner, resume jawaban responden, dan dokumentasi kuesioner.
2. Spesifikasi Aplikasi atau Sistem yang dikembangkan meliputi spesifikasi teknis aplikasi, tautan unduh aplikasi, manual penggunaan aplikasi, hingga screenshot aplikasi.
3. Cuplikan kode yang sekiranya penting dan ditambahkan.
4. Tabel yang terlalu panjang yang masih diperlukan tetapi tidak memungkinkan untuk ditayangkan di bagian utama skripsi.
5. Gambar-gambar pendukung yang tidak terlalu penting untuk ditampilkan di bagian utama. Akan tetapi, mendukung argumentasi/pengamatan/analisis.
6. Penurunan rumus-rumus atau pembuktian suatu teorema yang terlalu panjang dan terlalu teknis sehingga Anda berasumsi bahwa pembaca biasa tidak akan menelaah lebih lanjut. Hal ini digunakan untuk memberikan kesempatan bagi pembaca tingkat lanjut untuk melihat proses penurunan rumus-rumus ini.

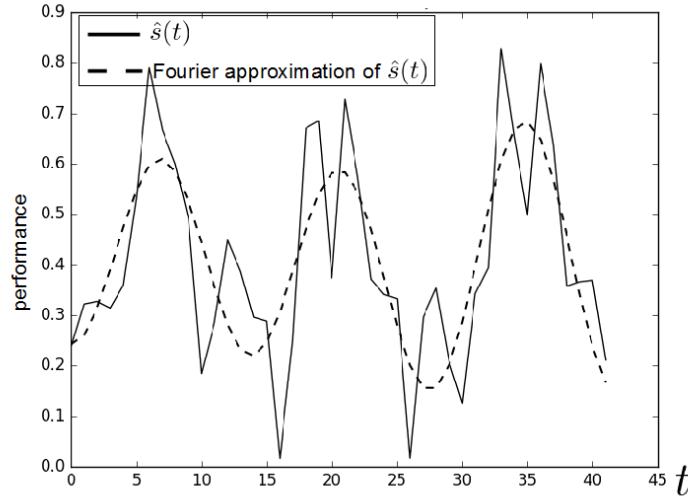
## LAMPIRAN

### L.2 Panduan Latex

#### L.2.1 Syntax Dasar

##### L.2.1.1 Penggunaan Sitasi

##### L.2.1.2 Penulisan Gambar



Gambar 1. Contoh gambar.

##### L.2.1.3 Penulisan Tabel

Tabel 1. Tabel ini

ID	Tinggi Badan (cm)	Berat Badan (kg)
A23	173	62
A25	185	78
A10	162	70

Contoh penulisan tabel bisa dilihat pada Tabel 1.

##### L.2.1.4 Penulisan formula

Contoh penulisan formula

$$L_{\psi_z} = \{t_i \mid v_z(t_i) \leq \psi_z\} \quad (1)$$

Contoh penulisan secara *inline*:  $PV = nRT$ . Untuk kasus-kasus tertentu, kita membutuhkan perintah "mathit" dalam penulisan formula untuk menghindari adanya jeda saat penulisan formula.

Contoh formula **tanpa** menggunakan "mathit":  $PVA = RTD$

Contoh formula **dengan** menggunakan "mathit":  $PVA = RTD$

### L.2.1.5 Contoh list

Berikut contoh penggunaan list

1. First item
2. Second item
3. Third item

## L.2.2 Blok Beda Halaman

### L.2.2.1 Membuat algoritma terpisah

Untuk membuat algoritma terpisah seperti pada contoh berikut, kita dapat memanfaatkan perintah *algstore* dan *algrestore* yang terdapat pada paket *algcompatible*. Pada dasarnya, kita membuat dua blok algoritma dimana blok pertama kita simpan menggunakan *algstore* dan kemudian di-restore menggunakan *algrestore* pada algoritma kedua. Perintah tersebut dimaksudkan agar terdapat kesinambungan antara kedua blok yang sejatinya adalah satu blok.

---

#### Algorithm 1 Contoh algorima

---

```
1: procedure CREATESET( $v$ )
2:   Create new set containing  $v$ 
3: end procedure
```

---

Pada blok algoritma kedua, tidak perlu ditambahkan caption dan label, karena sudah menjadi satu bagian dalam blok pertama. Pembagian algoritma menjadi dua bagian ini berguna jika kita ingin menjelaskan bagian-bagian dari sebuah algoritma, maupun untuk memisah algoritma panjang dalam beberapa halaman.

---

```
4: procedure CONCATSET( $v$ )
5:   Create new set containing  $v$ 
6: end procedure
```

---

### L.2.2.2 Membuat tabel terpisah

Untuk membuat tabel panjang yang melebihi satu halaman, kita dapat mengganti kombinasi *table* + *tabular* menjadi *longtable* dengan contoh sebagai berikut.

Tabel 2. Contoh tabel panjang

header 1	header 2
foo	bar

### L.2.2.3 Menulis formula terpisah halaman

Terkadang kita butuh untuk menuliskan rangkaian formula dalam jumlah besar sehingga melewati batas satu halaman. Solusi yang digunakan bisa saja dengan memindahkan satu blok formula tersebut pada halaman yang baru atau memisah rangkaian formula menjadi dua bagian untuk masing-masing halaman. Cara yang pertama mungkin akan menghasilkan alur yang berbeda karena ruang kosong pada halaman pertama akan diisi oleh teks selanjutnya. Sehingga di sini kita dapat memanfaatkan *align* yang sudah diatur dengan mode *allowdisplaybreaks*. Penggunaan *align* ini memungkinkan satu rangkaian formula terpisah berbeda halaman.

Contoh sederhana dapat digambarkan sebagai berikut.

$$\begin{aligned}
 x &= y^2 \\
 x &= y^3 \\
 a + b &= c \\
 x &= y - 2 \\
 a + b &= d + e \\
 x^2 + 3 &= y \\
 a(x) &= 2x
 \end{aligned} \tag{2}$$

$$b_i=5x$$

$$10x^2 = 9x$$

$$2x^2+3x+2=0$$

$$5x - 2 = 0$$

$$d=\log x$$

$$y=\sin x$$

## LAMPIRAN

### L.3 Format Penulisan Referensi

Penulisan referensi mengikuti aturan standar yang sudah ditentukan. Untuk internasionalisasi DTETI, maka penulisan referensi akan mengikuti standar yang ditetapkan oleh IEEE (*International Electronics and Electrical Engineers*). Aturan penulisan ini bisa diunduh di <http://www.ieee.org/documents/ieeecitationref.pdf>. Gunakan Mendeley sebagai *reference manager* dan *export* data ke format Bibtex untuk digunakan di Latex.

Berikut ini adalah sampel penulisan dalam format IEEE:

#### L.3.1 Book

##### Basic Format:

[1] J. K. Author, “Title of chapter in the book,” in Title of His Published Book, xth ed. City of Publisher, Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.

##### Examples:

- [1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.
- [2] L. Stein, “Random patterns,” in Computers and You, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.
- [3] R. L. Myer, “Parametric oscillators and nonlinear materials,” in Nonlinear Optics, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.
- [4] M. Abramowitz and I. A. Stegun, Eds., Handbook of Mathematical Functions (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.
- [5] E. F. Moore, “Gedanken-experiments on sequential machines,” in Automata Studies (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.
- [6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), Integrated Electronic Systems. Englewood Cliffs, NJ: Prentice-Hall, 1970.
- [7] M. Gorkii, “Optimal design,” Dokl. Akad. Nauk SSSR, vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., The Mathematical Theory of Optimal Processes. New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).
- [8] G. O. Young, “Synthetic structure of industrial plastics,” in Plastics, vol. 3,

Polymers of Hexadromicon, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64.

### **L.3.2 Handbook**

#### **Basic Format:**

- [1] Name of Manual/Handbook, x ed., Abbrev. Name of Co., City of Co., Abbrev. State, year, pp. xx-xx.

#### **Examples:**

- [1] Transmission Systems for Communications, 3rd ed., Western Electric Co., Winston Salem, NC, 1985, pp. 44-60.
- [2] Motorola Semiconductor Data Manual, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.
- [3] RCA Receiving Tube Manual, Radio Corp. of America, Electronic Components and Devices, Harrison, NJ, Tech. Ser. RC-23, 1992.

### **Conference/Prosiding**

#### **Basic Format:**

- [1] J. K. Author, "Title of paper," in Unabbreviated Name of Conf., City of Conf., Abbrev. State (if given), year, pp.xxx-xxx.

#### **Examples:**

- [1] J. K. Author [two authors: J. K. Author and A. N. Writer ] [three or more authors: J. K. Author et al.], "Title of Article," in [Title of Conf. Record as ], [copyright year] © [IEEE or applicable copyright holder of the Conference Record]. doi: [DOI number]

### **Sumber Online/Internet**

#### **Basic Format:**

- [1] J. K. Author. (year, month day). Title (edition) [Type of medium]. Available: [http://www.\(URL\)](http://www.(URL))

#### **Examples:**

- [1] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: [ht-  
tp://www.atm.com](http://www.atm.com)

### **Skripsi, Tesis dan Disertasi**

#### **Basic Format:**

- [1] J. K. Author, "Title of thesis," M.S. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

[2] J. K. Author, "Title of dissertation," Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

**Examples:**

[1] J. O. Williams, "Narrow-band analyzer," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993. [2] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993

## LAMPIRAN

### L.4 Contoh Source Code

#### L.4.1 Sample algorithm

---

##### Algorithm 2 Kruskal's Algorithm

---

```
1: procedure MAKESET( $v$ )
2:     Create new set containing  $v$ 
3: end procedure
4:
5: function FINDSET( $v$ )
6:     return a set containing  $v$ 
7: end function
8:
9: procedure UNION( $u,v$ )
10:    Unites the set that contain  $u$  and  $v$  into a new set
11: end procedure
12:
13: function KRUSKAL( $V, E, w$ )
14:     $A \leftarrow \{\}$ 
15:    for each vertex  $v$  in  $V$  do
16:        MakeSet( $v$ )
17:    end for
18:    Arrange  $E$  in increasing costs, ordered by  $w$ 
19:    for each  $(u,v)$  taken from the sorted list do
20:        if FindSet( $u$ )  $\neq$  FindSet( $v$ ) then
21:             $A \leftarrow A \cup \{(u, v)\}$ 
22:            Union( $u, v$ )
23:        end if
24:    end for
25:    return  $A$ 
26: end function
```

---

## L.4.2 Sample Python code

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22    if M is None:
23        M = np.copy(VT)
24    else:
25        M = np.concatenate((M, VT), 1)
26
27    VT = np.zeros((n*m,1), int)
28
29    return M
```

### L.4.3 Sample Matlab code

```
1 function X = BitXorMatrix(A,B)
2 %function to compute the sum without charge of two vectors
3
4 %convert elements into usigned integers
5 A = uint8(A);
6 B = uint8(B);
7
8 m1 = length(A);
9 m2 = length(B);
10 X = uint8(zeros(m1, m2));
11 for n1=1:m1
12     for n2=1:m2
13         X(n1, n2) = bitxor(A(n1), B(n2));
14     end
15 end
```