

**UNIwersYTET RZESZOWSKI**  
**WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH**  
**INSTYTUT INFORMATYKI**



*Gabriel Łasicki*  
gl131465

*Informatyka*

*Sprawozdanie z ćwiczeń laboratoryjnych z przedmiotu*  
*Biometryczne Systemy Zabezpieczeń*

Sprawozdania  
Grupa: 02  
Lab nr: 03

Prowadzący  
dr Zbigniew Gomółka

Rzeszów 2025



# Spis treści

<b>1. Wprowadzenie</b>	7
1.1. Cele ćwiczenia	7
1.1.1. Jakiś tytuł	7
1.1.2. Jakiś tytuł 2	7
1.2. Zawartość pracy	7
<b>2. Pierwszy dokument</b>	8
2.1. Struktura dokumentu	8
2.2. Kompilacja	8
2.3. Narzędzia	9
2.4. Przygotowanie dokumentu	10
2.5. Sprawdzanie pisowni w VSC	10
<b>3. Przykłady używania podstawowych elementów w <math>\text{\LaTeX}</math></b>	12
3.1. Punktory i listy wyliczeniowe	12
3.2. Liczba	12
3.3. Rysunki, tabele, schematy, wykresy	12
3.3.1. Rysunki	13
3.3.2. Tabele	14
3.4. Wzory matematyczne	15
3.5. Code Listing	15
3.5.1. Style i kolory kodu	16
3.6. Cytowania literatury	19
<b>4. Struktura sprawozdania z ćwiczeń laboratoryjnych - Biometryczne systemy zabezpieczeń</b>	20
4.1. Instrukcja przygotowania sprawozdania	20
<b>5. Przykład sprawozdania</b>	22
5.1. Cel ćwiczenia	22
5.2. Wstęp teoretyczny	22
5.3. Opis wykorzystywanego oprogramowania i narzędzi	22
5.4. Przebieg eksperymentu	23
5.4.1. Wczytanie obrazu i analiza histogramu	23
5.4.2. Wyrównanie histogramu	23
5.5. Wyniki i analiza	24
5.6. Wnioski	24
5.7. Bibliografia	24
<b>Bibliografia</b>	25
<b>Spis rysunków</b>	26
<b>Spis tabel</b>	27

<b>Spis listingów .....</b>	<b>28</b>
<b>Oświadczenie studenta o samodzielności pracy .....</b>	<b>29</b>

# 1. Wprowadzenie

L<sup>A</sup>T<sub>E</sub>X jest systemem składu umożliwiającym tworzenie dowolnego typu dokumentów (w szczególności naukowych i technicznych) o wysokiej jakości typograficznej [6, 8]. Wysoka jakość składu jest niezależna od rozmiaru dokumentu – zaczynając od krótkich listów do bardzo grubych książek. L<sup>A</sup>T<sub>E</sub>X automatyzuje wiele prac związanych ze składaniem dokumentów np.: referencje, cytowania, generowanie spisów (treści, rysunków, symboli itp.) itd.

L<sup>A</sup>T<sub>E</sub>X jest zestawem instrukcji umożliwiającym autorom skład i wydruk ich prac na najwyższym poziomie typograficznym. Do formatowania dokumentu L<sup>A</sup>T<sub>E</sub>X stosuje T<sub>E</sub>Xa (wymawiamy 'tech' – greckie litery  $\tau$ ,  $\epsilon$ ,  $\chi$ ). Korzystając z systemu składu L<sup>A</sup>T<sub>E</sub>X mamy za zadanie przygotować jedynie tekst źródłowy, cały ciężar składania, formatowania dokumentu przejmuje na siebie system.

## 1.1. Cele ćwiczenia

Celem poniższej pracy jest zapoznanie studentów z systemem L<sup>A</sup>T<sub>E</sub>X w zakresie umożliwiającym im samodzielne, profesjonalne złożenie pracy dyplomowej w systemie L<sup>A</sup>T<sub>E</sub>X.

### 1.1.1. Jakiś tytuł

#### 1.1.1.1. Jakiś tytuł w subsection

### 1.1.2. Jakiś tytuł 2

## 1.2. Zawartość pracy

W rozdziale 2 przedstawiono podstawowe informacje dotyczące struktury dokumentów w L<sup>A</sup>T<sub>E</sub>Xu.

## 2. Pierwszy dokument

W rozdziale tym przedstawiono podstawowe informacje dotyczące struktury prostych plików  $\text{\LaTeX}$  a. Omówiono również metody kompilacji plików z zastosowaniem programów *latex* oraz *pdflatex*.

### 2.1. Struktura dokumentu

Plik  $\text{\LaTeX}$  owy jest plikiem tekstowym, który oprócz tekstu zawiera polecenia formatujące ten tekst (analogicznie do języka HTML). Plik składa się z dwóch części:

1. Preambuły – określającej klasę dokumentu oraz zawierającej m.in. polecenia dołączającej dodatkowe pakiety;
2. Części głównej – zawierającej zasadniczą treść dokumentu.

```
\documentclass[a4paper,12pt]{article}      % preambuła
\usepackage[polish]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{times}

\begin{document}                          % część główna

\section{Sztuczne życie}

% treść
% ąśęźźćńłóĘŚĄŻŻĆŃÓŁ

\end{document}
```

Nie ma żadnych przeciwwskazań do tworzenia dokumentów w  $\text{\LaTeX}$  u w języku polskim. Plik źródłowy jest zwykłym plikiem tekstowym i do jego przygotowania można użyć dowolnego edytora tekstów, a polskie znaki wprowadzać używając prawego klawisza **Alt**. Jeżeli po kompilacji dokumentu polskie znaki nie są wyświetlane poprawnie, to na 95% źle określono sposób kodowania znaków (należy zmienić opcje wykorzystywanych pakietów).

### 2.2. Kompilacja

Założmy, że przygotowany przez nas dokument zapisany jest w pliku `test.tex`. Kolejno wykonane poniższe polecenia (pod warunkiem, że w pierwszym przypadku nie wykryto błędów i kompilacja zakończyła się sukcesem) pozwalają uzyskać nasz dokument w formacie pdf:

```
latex test.tex
dvips test.dvi -o test.ps
ps2pdf test.ps
```

lub za pomocą  $\text{\LaTeX}$ :

```
pdflatex test.tex
```

Przy pierwszej kompilacji po zmianie tekstu, dodaniu nowych etykiet itp.,  $\text{\LaTeX}$  tworzy sobie spis rozdziałów, obrazków, tabel itp., a dopiero przy następnej kompilacji korzysta z tych informacji.

W pierwszym przypadku rysunki powinny być przygotowane w formacie eps, a w drugim w formacie pdf. Ponadto, jeżeli używamy polecenia `pdflatex test.tex` można wstawiać grafikę bitową (np. w formacie jpg).

## 2.3. Narzędzia

Istnieje wiele narzędzi do pisania w  $\text{\LaTeX}$ , które mogą pomóc w tworzeniu dokumentów i zarządzaniu nimi. Oto lista popularnych narzędzi i edytorów  $\text{\LaTeX}$ , które są dostępne zarówno w wersji online, jak i offline.

### Narzędzia online do pisania w $\text{\LaTeX}$ .

- Overleaf – jeden z najpopularniejszych edytorów  $\text{\LaTeX}$  online. Umożliwia współpracę z innymi użytkownikami w czasie rzeczywistym oraz zapewnia automatyczną kompilację dokumentów. Overleaf oferuje szereg gotowych szablonów (np. prace dyplomowe, CV, artykuły naukowe). Wśród funkcji można wyróżnić m.in.: kompilacja  $\text{\LaTeX}$  w czasie rzeczywistym, wsparcie dla XeLaTeX, LuaLaTeX, BibTeX, Biber, współpraca z wieloma użytkownikami, gotowe szablony oraz integracja z GitHub. Szczegółowe informacje dostępne są pod adresem: <https://www.overleaf.com/>.
- Papeeria – edytor  $\text{\LaTeX}$  online podobny do Overleaf. Umożliwia współpracę w czasie rzeczywistym i ma integrację z GitHub. Oferuje bezpłatną wersję z podstawowymi funkcjami. Funkcje: wsparcie dla współpracy online, kompilacja  $\text{\LaTeX}$  w chmurze. Szczegółowe informacje dostępne są pod adresem: <https://www.papeeria.com/>.

### Edytory offline do pisania w LaTeX.

- Visual Studio Code (VSC) – to potężne narzędzie do pisania dokumentów w  $\text{\LaTeX}$ , szczególnie po zainstalowaniu rozszerzenia  $\text{\LaTeX}$  Workshop. Zapewnia ono pełne wsparcie dla kompilacji, podglądu PDF, inteligentnego uzupełniania składni oraz współpracy z narzędziami bibliograficznymi jak BibTeX i Biber. Dzięki integracji z systemem zarządzania plikami i narzędziami do wersjonowania (jak Git), VS Code staje się bardzo elastycznym narzędziem do pisania profesjonalnych dokumentów, artykułów naukowych i prac dyplomowych w  $\text{\LaTeX}$ . Szczegółowe informacje odnośnie konfiguracji środowiska do pisania LaTeX w VSC można znaleźć a instrukcji "*How to write LaTeX documents using Visual Studio Code*" pod adresem <https://www.geekering.com/programming-languages/filipesalgueiro/how-to-write-latex-documents-using-visual-studio-code/>
- TeXstudio – to zintegrowane środowisko do tworzenia dokumentów  $\text{\LaTeX}$ . Celem jest uczynienie pisania w  $\text{\LaTeX}$  u tak łatwym i wygodnym, jak to tylko możliwe. Dlatego TeXstudio ma wiele funkcji, takich jak podświetlanie składni, zintegrowana przeglądarka, sprawdzanie referencji i różni asystenci. Więcej szczegółów można znaleźć w funkcjach. Szczegółowe informacje dostępne są pod adresem: <https://www.texstudio.org/>

### Dodatkowe narzędzia wspierające pisanie w LaTeX.

- Mendeley – popularne narzędzie do zarządzania bibliografią i organizacji dokumentów naukowych. Obsługuje format BibTeX i pozwala na łatwą integrację z LaTeX. Szczegółowe informacje dostępne są pod adresem: <https://www.mendeley.com>.
- JabRef – darmowe narzędzie do zarządzania bibliografią i referencjami, które wspiera format BibTeX i BibLaTeX. Ułatwia organizowanie źródeł bibliograficznych i integrację z LaTeX. Zobacz: <https://www.jabref.org/>.

- Zotero – kolejne narzędzie do zarządzania referencjami naukowymi, które wspiera BibTeX i BibLaTeX. Zotero jest popularnym wyborem wśród studentów i badaczy. Szczegółowe informacje dostępne są pod adresem: <https://www.zotero.org>.

Wśród środowisk online wyróżnić można: Overleaf, Papeeria i Authorea oferują wygodną współpracę i szybki dostęp do narzędzi LaTeX z dowolnego miejsca. Środowiska offline: TeXmaker, TeXworks, Kile i LyX to doskonałe edytory dla pracy lokalnej z rozbudowanymi funkcjami. Dodatkowe narzędzia: JabRef, Mendeley i Zotero ułatwiają zarządzanie bibliografią i cytowaniami w LaTeX. Wybór odpowiedniego narzędzia zależy od Twoich potrzeb, czy preferujesz pracę lokalną, czy online, a także jak bardzo zaawansowane funkcje są Ci potrzebne.

## 2.4. Przygotowanie dokumentu

Plik źródłowy LaTeX jest zwykłym plikiem tekstowym. Przygotowując plik źródłowy warto wiedzieć o kilku szczegółach:

- Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja. Aby uzyskać *twardą spację*, zamiast znaku spacji należy użyć znaku *tyldy*.
- Znakiem końca akapitu jest pusta linia (ilość pustych linii nie ma znaczenia), a nie znaki przejścia do nowej linii.
- LaTeX sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.

Przydatna pomoc do pracy w LaTeX:

- Dokumentacja dostępna w Overleaf <https://www.overleaf.com/learn>.
- Edytor równań dla matematyki online <https://editor.codecogs.com/>.
- Edytor/generator tabel <https://www.tablesgenerator.com/>.

## 2.5. Sprawdzanie pisowni w VSC

Visual Studio Code można sprawdzać pisownię w dokumentach LaTeX, w tym także w języku polskim. W tym celu należy zainstalować poniższe rozszerzenia.

1. Zainstalowanie rozszerzenia "Code Spell Checker To rozszerzenie (streetsidesoftware.code-spell-checker) umożliwia sprawdzanie pisowni w różnych językach. Poniżej kroki jak dodać obsługę języka polskiego:
  - Otwórz Visual Studio Code.
  - Przejdź do Extensions (Ctrl + Shift + X).
  - Wyszukaj i zainstaluj Code Spell Checker.
  - Następnie zainstaluj rozszerzenie Polish - Code Spell Checker (streetsidesoftware.code-spell-checker-polish). Po instalacji sprawdzanie pisowni dla języka polskiego powinno działać
2. Dodanie języka polskiego do konfiguracji - jeśli chcesz, aby sprawdzanie pisowni działało domyślnie w języku polskim, dodaj wpis do ustawień:



- Otwórz Ustawienia (Ctrl + ,).
  - Wyszukaj "cSpell.language".
  - Kliknij "Edit in settings.json".
  - Dodaj lub edytuj ustawienie: "cSpell.language": "pl,en". Możesz też dodać pl do listy obsługiwanych języków, jeśli już masz tam en.
3. Sprawdzanie pisowni w czasie rzeczywistym - Po instalacji rozszerzenia błędnie napisane słowa będą podkreślane na czerwono. Możesz kliknąć na podkreślone słowo i wybrać poprawną sugestię.

## 3. Przykłady używania podstawowych elementów w L<sup>A</sup>T<sub>E</sub>X

### 3.1. Punktory i listy wyliczeniowe

Listy punktowane wyglądają następująco:

- First bullet;
- Second bullet;
- Third bullet.

Listy numerowane można dodawać w następujący sposób:

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

Tekst jest kontynuowany tutaj.

### 3.2. Liczba

Pakiet `siunitx` zadba o to, by liczba została poprawnie sformatowana:

1 234 567 890,098 765 432 1

### 3.3. Rysunki, tabele, schematy, wykresy

Wszystkie rysunki i tabele powinny być cytowane w tekście głównym jako Rys. 3.1, Tabela 3.1 itd. W tekście najpierw pojawia się opis do rysunku/tabeli oraz odwołanie do elementu, a następnie rysunek bądź tabela. Rysunki, wykresy, schematy mają być w formacie \*.eps, nazwy rysunków odpowiednio numerowane zgodnie z założeniem fig\_0001, fig\_0002 itd. Rysunki muszą być czytelne, dlatego należy zastosować wysoką jakość rysunków, które muszą być dostarczone w odpowiednio wysokiej rozdzielczości (min. 1000 pikseli szerokości/wysokości lub rozdzielczość 300 dpi lub wyższa). W celu przekonwertowania rysunków do pliku eps można użyć darmowego konwertera plików online ONLINE-CONVERT <https://www.online-convert.com/>. W przypadku gdy rysunek nie mieści się na danej stronie należy użyć polecenia `\newpage` lub `\clearpage` co pokazano w poniższym przykładzie.

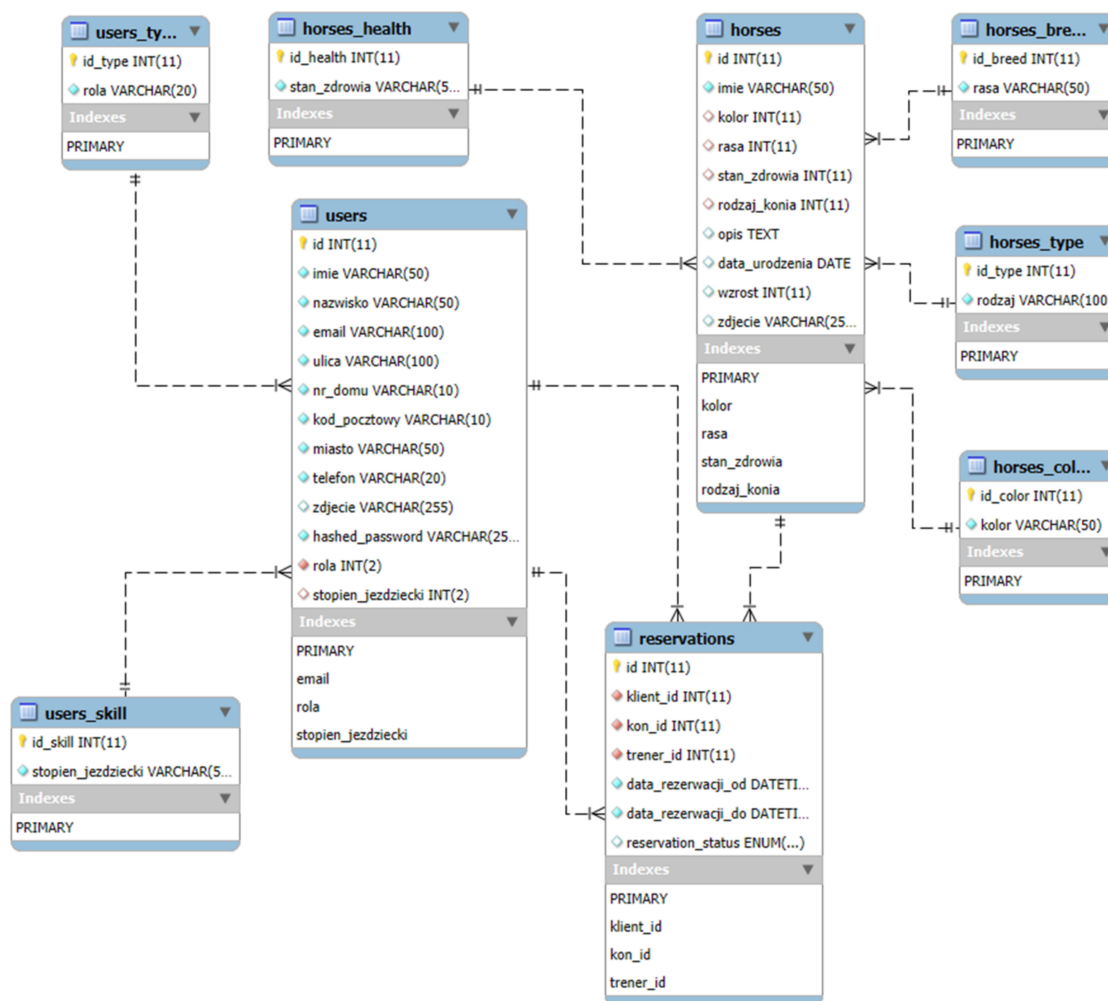
**UWAGA!!!**

Do wszystkich rysunków/tabel należy umieścić źródła tj. np. rysunki w postaci wektorowej, zestaw danych który był używany do wygenerowania wykresów itp. O szczegóły należy dopytać promotora pracy.

Schematy można przygotować z użyciem narzędzia [diagrams.net/draw](https://app.diagrams.net/) io w wersji online lub desktopowej <https://app.diagrams.net/> oraz w środowisku Microsoft Visio <https://www.microsoft.com/pl-pl/microsoft-365/visio/flowchart-software>.

### 3.3.1. Rysunki

W systemie zarządzania stadniną koni, dane będą przechowywane i zarządzane za pomocą relacyjnej bazy danych MySQL. Struktura bazy danych została zaprojektowana w taki sposób, aby umożliwić łatwe zarządzanie informacjami o koniach, rezerwacjach, klientach i trenerach. Baza danych składa się z kilku głównych tabel, takich jak: `horses`, `clients`, `trainers`, `reservations` oraz tabel pomocniczych: `horses_breed`, `horses_color`, `health_status`, `horse_type`, które przechowują szczegółowe informacje o różnych aspektach zarządzania stadniną. Diagram ERD przedstawiono na Rys. 3.1.



Rys. 3.1. Diagram ERD projektowanej aplikacji.

Pakiet `subcaption` pozwala na umieszczanie w podpisie rysunku odnośników do „podilustracji”:



**Rys. 3.2.** Przykład użycia `\subcaption`: (a) litera A, (b) litera B.

### 3.3.2. Tabele

Poniżej przykład tworzenia tabeli patrz Tabela 3.1 i Tabela 3.2.

**Tabela 3.1.** To jest tytuł tabeli.

Name	Type	Activations	Learnables	
Name 1	Type 1	232	–	
Name 2	Type 1	512	InputWeights	2048x232
			RecurrentWeights	2048x512
			Bias	2048x1
Name 3	Type 1	8/16/32	Weights	8/16/32x512
			Bias	8/16/32x32
Name 4	Type 1	8/16/32	–	
Name 5	Type 1	8/16/32	–	

**Tabela 3.2.** Klasy trajektorii zadań

Lp	UAV	Czas [s]
1	36 × 6	8778
2	36 × 6	8743
3	36 × 6	8765
4	36 × 6	8882
5	36 × 6	8750
6	36 × 6	8965
7	36 × 6	8727
8	36 × 6	8937
9	36 × 6	8750

### 3.4. Wzory matematyczne

Kryterium  $Q$  jest separowalne dla procesu  $P$ , jeżeli dla każdej sekwencji decyzyjnej  $\tilde{u} \in U$  można w rekurencyjny sposób wyznaczyć:

$$Q_0 = \text{const w szczególności gdy } Q_0 = 0 \quad (3.1)$$

$$Q_{i+1} = f_Q(Q_i, u_i, si) \text{ dla } i = 0, 1, \dots, d(\tilde{s} - 1) \quad (3.2)$$

gdzie:  $Q_i$  dla  $f > 0$  opisuje wartość kryterium  $Q$  wyznaczonego dla tego stanu w rozpatrywanej trajektorii  $Q_i = Q(\tilde{u}')$ , gdzie  $\tilde{u}' = (u_0, u_1, \dots, u_{i-1})$  jest początkową częścią zbioru  $\tilde{u}'$ . Funkcją  $f_Q : \mathfrak{R} \times U \times S \rightarrow \mathfrak{R}$  jest częściową funkcją, taką że:

$$\text{Dom } f_Q = \{(a, u, s) \mid \mathfrak{R} \times U \times S : s \in S^P, u \in U_p, a \in \mathfrak{R}\} \quad (3.3)$$

Poniżej zamieszczono przykład umieszczenia długich wzorów w pracy. Zbiór stanów niedopuszczalnych:

$$S_N = \{(x, t) : (J_M(x, t) \cap J_{out} \neq 0 \wedge F^r < t < B^{r+1}, 1 \leq r \leq ILs - 1) \cup \left( \bigcup_{j \in J_{out}} \exists j \notin x^0 \wedge t > F^{ILs} \right) \cup \left( \bigcup_{j \in J_{baz}} \exists j \notin x^0 \wedge t > T_{max} \right)\} \quad (3.4)$$

gdzie  $T_{max}$  – przyjęty czas zakończenia wszystkich zadań w danym dniu.

Czasem zachodzi potrzeba wytłumaczenia znaczenia symboli użytych w równaniu. Można to zrobić z użyciem zdefiniowanego na potrzeby niniejszej klasy środowiska eqwhere.

$$E = mc^2 \quad (3.5)$$

gdzie

$m$  – masa

$c$  – prędkość światła w próżni

Odległość półpauzy od lewego marginesu należy dobrać pod kątem najdłuższego symbolu (bądź listy symboli) poprzez odpowiednie ustawienie parametru tego środowiska (domyślnie: 2 cm).

### 3.5. Code Listing

Kod jest zwykle przechowywany w pliku źródłowym, dlatego polecenie, które automatycznie pobiera kod z pliku, staje się bardzo przydatne. W pracy należy umieszczać kody źródłowe które przechowywane będą w katalogu `src` w oddzielnych plikach z odpowiednim rozszerzeniem np. `*.m`, `*.cpp`, `*.css`, `*.py`, `*.html`, `*.css` itp. Poniżej przedstawiono przykład jak dodawać kod źródłowy z pliku, więcej informacji można znaleźć w dokumentacji Code listing [L<sup>A</sup>T<sub>E</sub>X](https://www.overleaf.com/learn/latex/Code_listing) [https://www.overleaf.com/learn/latex/Code\\_listing](https://www.overleaf.com/learn/latex/Code_listing).

Algorytm Euklidesa służy do wyznaczania największego wspólnego dzielnika dwóch liczb całkowitych. Największy wspólny dzielnik dwóch liczb  $a$  i  $b$ , to taka liczba, która dzieli te liczby bez reszty i jest ona możliwie największa. Można go zastosować do skracania ułamków lub wyznaczenia najmniejszej wspólnej wielokrotności NWW. Artykuł opisuje dwie postacie algorytmu: nieoptymalną i optymalną.

- Nieoptymalizowany algorytm Euklidesa – sposób rozwiązania jest następujący: wybieramy większą z dwóch liczb i zamieniamy ją na różnicę większej i mniejszej. Czynność tą powtarzamy do momentu uzyskania dwóch takich samych wartości. Warto zaznaczyć, że ten algorytm jest bardzo niewydajny. Gdy dobierzemy odpowiednio liczby, to ilość operacji znacznie się zwiększy.
- Zoptymalizowany algorytm Euklidesa – w przypadku optymalnego rozwiązania NWD postępujemy następująco: założymy, że wyznaczamy NWD dwóch liczb naturalnych  $a$  i  $b$ . W każdym przejściu pętli wykonujemy dwie operacje:  $a = b$  oraz  $b = a \bmod b$ . Czynności te powtarzamy do momentu, gdy zmienna  $b$  osiągnie wartość zero. Zmienna  $a$  będzie przechowywać wtedy największy wspólny dzielnik liczb podanych na wejściu.

Przykładowa implementacja algorytmu NWD w wersji rekurencyjnej dla powyższych wersji algorytmu przedstawiono na Listing 3.1 i 3.2.

**Listing 3.1.** Nieoptymalizowany algorytm NWD wersja rekurencyjna

```

1 # *****
2 # Nieoptymalizowany algorytm Euklidesa - wersja rekurencyjna
3 # Dane wejściowe: a, b
4 # Dane wyjściowe: a
5 # *****
6 def NWD(a,b):
7     if a!=b:
8         if a>b: return NWD(a-b,b)
9         else: return NWD(a,b-a)
10    return a

```

**Listing 3.2.** Zoptymalizowany algorytm NWD wersja rekurencyjna

```

1 # *****
2 # Zoptymalizowany algorytm Euklidesa - wersja rekurencyjna
3 # Dane wejściowe: a, b
4 # Dane wyjściowe: a
5 # *****
6 def nwdrek(a,b):
7     if b!=0:
8         return nwdrek(b, a % b)
9     return a

```

### 3.5.1. Style i kolory kodu

Formatowanie kodu w pakiecie listingowym jest wysoce konfigurowalne. W przypadku gdy chcemy aby kod wstawiany za pomocą listings w  $\text{\LaTeX}$  był kolorowy, należy skonfigurować styl, który umożliwi kolorowanie składni. Do tego potrzebna jest paczka `xcolor`, która pozwala na definiowanie kolorów. Poniżej kroki, jak to zrobić:

- W preambule dodaj odpowiednie paczki: `\usepackagelistings`, `\usepackagecolor`.
- Następnie zdefiniuj styl dla kodu źródłowego. Na przykład, dla języka Python może wyglądać następująco:

```

\lstdefinestyle{mystyle}{
    commentstyle=\color{green},          % Kolor komentarzy
    keywordstyle=\color{blue},           % Kolor słów kluczowych

```

```

        numberstyle=\tiny\color{gray},           % Kolor i styl numerów linii
        stringstyle=\color{red},                 % Kolor ciągów znaków
        basicstyle=\ttfamily\footnotesize,       % Podstawowy styl kodu
        breakatwhitespace=false,                 % Automatyczne dzielenie wierszy
        breaklines=true,                         % Dzielenie długich linii
        keepspaces=true,                         % Zachowanie spacji
        numbers=left,                             % Numery linii po lewej
        numbersep=5pt,                           % Odstęp numerów od kodu
        showspaces=false,                         % Nie pokazuj spacji
        showstringspaces=false,                  % Nie pokazuj spacji
                                                % w ciągach znaków
        showtabs=false,                           % Nie pokazuj tabulacji
        tabsize=2                                % Rozmiar tabulacji
    }

```

- Aby użyć tego stylu w kodzie, dodaj do `\lstinputlisting` argument `style=mystyle`. Cała komenda będzie wyglądać tak:

```

\lstinputlisting[caption=Algorytm NWD, style=mystyle]
{src/nwdRedVer1.py}

```

- Możesz też dostosować kolory według własnych preferencji, korzystając z paczki `xcolor` i np. definiując własne kolory:

```

\definecolor{lightgray}{rgb}{0.9,0.9,0.9}
\definecolor{blue}{rgb}{0,0,1}
\definecolor{green}{rgb}{0,0.6,0}
\definecolor{red}{rgb}{0.6,0,0}
\definecolor{gray}{rgb}{0.5,0.5,0.5}

```

Na Listing 3.3 przedstawiono uzyskany wynik użycia zdefiniowane powyżej stylu.

**Listing 3.3.** Nieoptymalizowany algorytm NWD wersja rekurencyjna

```

1 # *****
2 # Nieoptymalizowany algorytm Euklidesa - wersja rekurencyjna
3 # Dane wejściowe: a, b
4 # Dane wyjściowe: a
5 # *****
6 def NWD(a,b):
7     if a!=b:
8         if a>b: return NWD(a-b,b)
9         else: return NWD(a,b-a)
10    return a

```

Na Listing 3.4 przedstawiono przykładowy listing w JAVA.

**Listing 3.4.** Przykładowy listing w JAVA

```

1 package ExampleTask;
2
3 import java.util.Scanner;
4

```

```

5  /**
6   * 1. Napisz program, w którym zadeklarujesz tablicę 9-elementową. Wartości tablicy mają
       być wczytane z klawiatury
7   * (wprowadzone przez użytkownika). Wartości wyświetl na ekranie w jednym wierszu.
       Sprawdź czy wartość środkowego
8   * elementy czy (czwarty element tablicy) jest większy, mniejszy czy równy średniej
       arytmetycznej dwóch brzegowych
9   * wartości (pierwszego i ostatniego). Wyświetl odpowiedni komunikat.
10  */
11
12 public class Task1 {
13
14     public static void main(String[] args) {
15         run();
16
17     }
18
19     public static void run() {
20         int [] array = new int[9];
21         //wczytywanie danych z klawiatury
22         for (int i = 0; i < array.length; i++) {
23             array[i]=InputInt();
24         }
25
26         //wystwietlenie
27         for (int item: array ) {
28             System.out.print(item + " ");
29         }
30         System.out.print("\nWartosc środkowego elementu: "+ array[4]);
31         float average = (array[0]+array[8])/2;
32         if (array[4]>average) System.out.print(", jest większy od sredniej, która wynosi
"+ average);
33         else if (array[4]<average) System.out.print(", jest mniejszy od sredniej, która
wynosi "+ average);
34         else if (array[4]>=average) System.out.print(", jest równy sredniej, która
wynosi "+ average);
35     }
36
37     public static int InputInt(){
38         Scanner scanner = new Scanner(System.in);
39         System.out.println("Podaj liczbe: ");
40         int number = scanner.nextInt();
41         return number;
42     }
43 }

```

Na Listing 3.5 przedstawiono przykładowy listing w C#.

#### Listing 3.5. Przykładowy listing w C#

```

1 //main
2
3 Osoba osoba = new Osoba("Jan Nowak", 25);
4 osoba.View();
5
6
7 public class Osoba
8 {

```



```
9 public string Imie { get; set; }
10 public int Wiek { get; set; }
11
12 public Osoba(string imie, int wiek)
13 {
14     Imie = imie;
15     Wiek = wiek;
16 }
17 public void View()
18 {
19     System.Console.WriteLine($"Imię: {Imie}, wiek: {Wiek}");
20 }
21 }
```

### 3.6. Cytowania literatury

W LaTeX-u, `\cite` jest używane do cytowania źródeł z bibliografii. Cytowanie w LaTeX-u wymaga użycia systemu bibliograficznego, który może być oparty na pakiecie `bibtex` lub `biblatex`. Oto jak można używać `\cite` w różnych kontekstach:

Przygotowanie pliku bibliograficznego - stwórz plik `.bib` (np. `bibliografia.bib`) z wpisami bibliograficznymi. Każdy wpis powinien być sformatowany zgodnie z wymaganiami BibTeX. Przykład wpisu:

```
@article{smith2020,
  author = {John Smith},
  title = {A Great Paper},
  journal = {Journal of Great Papers},
  year = {2020},
  volume = {5},
  number = {3},
  pages = {123-456}
}
```

W pliku `.tex` załaduj pakiet `natbib` (opcjonalnie) i użyj `\cite`:

```
\documentclass{article}
\usepackage[numbers]{natbib} % Można również użyć [authoryear] lub innych opcji

\begin{document}

Here is a citation \cite{smith2020}.

\bibliographystyle{plain} % Wybierz odpowiedni styl bibliografii
\bibliography{bibliografia} % Plik bibliograficzny (bez rozszerzenia .bib)

\end{document}
```

W pracy można cytować literaturę pojedynco zgodnie z zapisem: [3]. Cytowanie kilku źródeł: [3, 4, 5, 9]. Kolejny przykład użycia cytowania [6, 8]. Przykład cytowania źródła internetowego [2].

## 4. Struktura sprawozdania z ćwiczeń laboratoryjnych - Biometryczne systemy zabezpieczeń

### 4.1. Instrukcja przygotowania sprawozdania

Sprawozdanie należy przygotować z wykorzystaniem udostępnionego szablonu  $\text{\LaTeX}$ . Dokument powinien być zgodny z wymaganym formatowaniem oraz spełniać określone kryteria merytoryczne. Każde sprawozdanie musi być wykonane oddzielnie dla każdego laboratorium.

Sprawozdanie powinno zawierać szczegółowe opracowanie zaproponowanych rozwiązań dla ćwiczeń oraz zadań do samodzielnego rozwiązania. W szczególności należy uwzględnić następujące elementy:

- Opis stosowanych algorytmów i metod - krótkie omówienie zastosowanych metod przetwarzania obrazów, w tym np. histogramowej segmentacji, binaryzacji, detekcji cech biometrycznych. Wskazanie ich znaczenia w analizie biometrycznej oraz opis działania.
- Opis wykorzystywanego oprogramowania i narzędzi - informacja o środowisku programistycznym (np. MATLAB + Image Processing Toolbox). Opis kluczowych funkcji wykorzystanych w ćwiczeniu. Kod źródłowy programu w postaci listingów umieszczonych zgodnie z wymaganiami dokumentacji technicznej.
- Przebieg eksperymentu - szczegółowy opis kolejnych etapów eksperymentu. Przygotowanie danych wejściowych, np. wczytanie obrazu, wstępne przetwarzanie. Realizacja poszczególnych operacji, np. segmentacja, binaryzacja, wykrywanie cech biometrycznych. Analiza wyników pośrednich, np. wykresy histogramów, zwizualizowane obrazy po przekształceniach.
- Wyniki i analiza - prezentacja wyników uzyskanych w trakcie eksperymentu. Porównanie różnych metod segmentacji i ich wpływu na jakość analizy biometrycznej. Analiza dokładności uzyskanych wyników oraz identyfikacja potencjalnych błędów.
- Wnioski - podsumowanie skuteczności zastosowanych metod. Możliwości poprawy wyników, np. poprzez zastosowanie bardziej zaawansowanych metod segmentacji. Potencjalne zastosowania w rzeczywistych systemach biometrycznych.
- Bibliografia - lista wykorzystanych materiałów, w tym podręczników, dokumentacji MATLAB-a, publikacji naukowych i innych źródeł.

Dokument należy sporządzić w sposób przejrzysty, stosując numerowane sekcje i podsekcje, zgodnie z zasadami formatowania w  $\text{\LaTeX}$ .

Praca powinna składać się z poniższych rozdziałów:

- Laboratorium 1
- Laboratorium 2
- Laboratorium 3
- ....

- Laboratorium n
- Bibliografia
- Spis rysunków
- Spis tabel
- Spis listingów
- Oświadczenie studenta o samodzielności pracy

## 5. Przykład sprawozdania

<b>Biometryczne Systemy Zabezpieczeń</b>	
<b>Ćwiczenia Laboratoryjne</b>	
Temat	
Ćwiczenie nr:	XX
Autorzy:	Gabriel Łasicki
Grupa laboratoryjna	XX
Zespół nr.	XX
Data wykonania ćwiczenia	XX.XX.XXXX
Data oddania ćwiczenia	XX.XX.XXXX

### 5.1. Cel ćwiczenia

Celem ćwiczenia jest analiza histogramu obrazu biometrycznego, jego zastosowanie w poprawie jakości obrazu oraz segmentacji. Zostaną wykorzystane operacje przetwarzania histogramu, takie jak wyrównywanie histogramu oraz binaryzacja obrazu na podstawie progowania.

### 5.2. Wstęp teoretyczny

Histogram obrazu to funkcja przedstawiająca rozkład wartości pikseli w obrazie. Może być wykorzystywany do:

- Poprawy kontrastu obrazu (rozciąganie histogramu, wyrównywanie histogramu),
- Segmentacji obrazu na podstawie progów jasności (np. metoda Otsu),
- Analizy jakości obrazu, w tym identyfikacji prześwietlonych lub niedoświetlonych obszarów.

Histogram w obrazie monochromatycznym można obliczyć za pomocą funkcji  $h(k)$ , określającej liczbę pikseli o poziomie jasności  $k$ :

$$h(k) = \sum_{(x,y) \in \Omega} \delta(I(x,y) - k), \quad (5.1)$$

gdzie  $\delta$  to funkcja Kroneckera, a  $\Omega$  oznacza zbiór pikseli w obrazie.

### 5.3. Opis wykorzystywanego oprogramowania i narzędzi

Do przeprowadzenia eksperymentu wykorzystano środowisko MATLAB oraz bibliotekę Image Processing Toolbox. Kluczowe funkcje wykorzystane w analizie to:

- `imhist` - obliczanie histogramu obrazu,
- `imadjust` - rozciąganie histogramu,
- `histeq` - wyrównywanie histogramu,
- `graythresh` oraz `imbinarize` - segmentacja metodą Otsu.

## 5.4. Przebieg eksperymentu

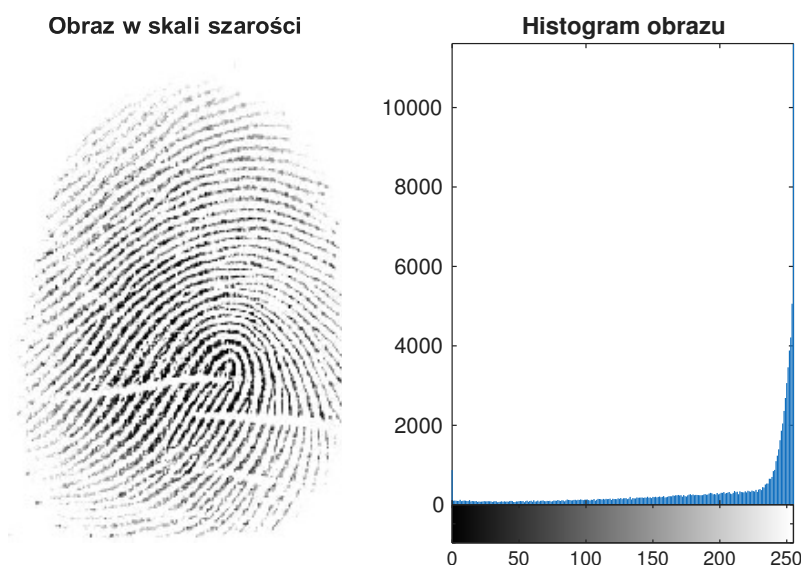
### 5.4.1. Wczytanie obrazu i analiza histogramu

Wczytano obraz w skali szarości oraz obliczono jego histogram, zgodnie Listing 5.1:

**Listing 5.1.** Wczytanie obrazu i obliczenie histogramu

```
1 % Wczytanie obrazu biometrycznego
2 I = imread('fingerprint.png');
3
4 % Sprawdzenie, czy obraz jest RGB czy grayscale
5 if size(I, 3) == 3
6     I_gray = rgb2gray(I); % Konwersja do skali szarości
7 else
8     I_gray = I; % Jeśli już jest w skali szarości, nie konwertujemy
9 end
10
11 % Wyświetlenie obrazu i jego histogramu
12 figure, imshow(I_gray), title('Obraz w skali szarości');
13 figure, imhist(I_gray), title('Histogram obrazu');
```

Na Rys. 5.1 przedstawiono otrzymany obraz i jego histogram.



**Rys. 5.1.** Obraz wejściowy i jego histogram.

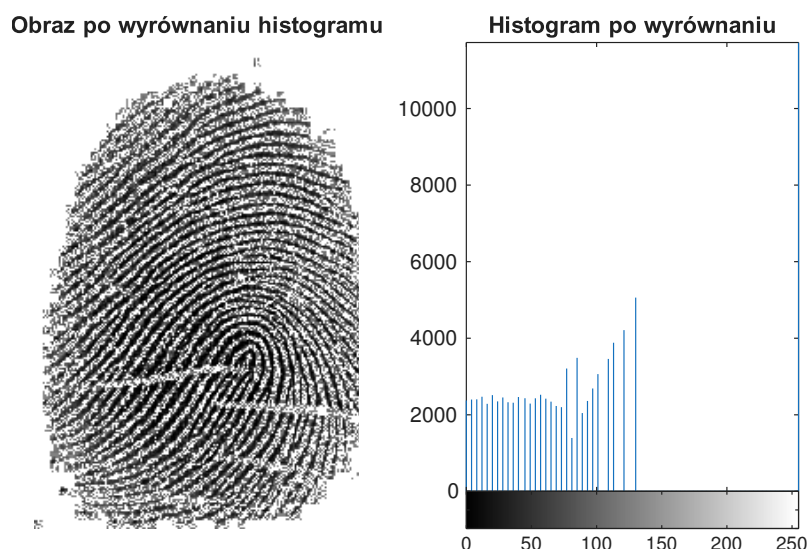
### 5.4.2. Wyrównanie histogramu

Przeprowadzono operację wyrównywania histogramu w celu poprawy kontrastu (patrz Listing 5.2):

**Listing 5.2.** Wyrównanie histogramu

```
1 I_eq = histeq(I_gray); % Wyrównanie histogramu
2
3 figure, imshow(I_eq), title('Obraz po wyrównaniu histogramu');
4 figure, imhist(I_eq), title('Histogram po wyrównaniu');
```

Na Rys. 5.2 przedstawiono otrzymane rezultaty po wyrównaniu histogramu.



Rys. 5.2. Obraz po wyrównaniu i jego histogram.

## 5.5. Wyniki i analiza

Po przeprowadzeniu analizy histogramu uzyskano następujące wyniki:

- Histogram oryginalnego obrazu wykazał skupienie wartości pikseli w wąskim zakresie, co wskazuje na niski kontrast.
- Po wyrównaniu histogramu kontrast obrazu uległ poprawie, a wartości pikseli zostały bardziej równomiernie rozłożone.
- Metoda Otsu pozwoliła na automatyczne wyznaczenie progu binaryzacji, co umożliwiło efektywne wydzielenie struktur biometrycznych na obrazie.

## 5.6. Wnioski

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że:

- Histogram jest przydatnym narzędziem w analizie jakości obrazu biometrycznego.
- Wyrównanie histogramu poprawia kontrast i może zwiększać czytelność obrazu.
- Segmentacja oparta na histogramie, w szczególności metoda Otsu, pozwala na skuteczne wydzielenie istotnych cech obrazu.

## 5.7. Bibliografia

- MATLAB Image Processing Toolbox Documentation [1],
- Rafael Gonzalez, Richard Woods, Digital Image Processing Global Edition [7].

# Bibliografia

- [1] Matlab image processing toolbox documentation, 2024. Accessed: 2025-02-15.
- [2] Title of the webpage, 2024. Accessed: 2024-09-16.
- [3] Ada Europe. *Ada Reference Manual ISO/IEC 8652:200y(E) Ed. 3*, 2006.
- [4] A. Burns and B. Dobbing. The Ravenscar Profile for real-time and high integrity systems. *CrossTalk*, 16(11):9–12, 2003.
- [5] A. Burns, B. Dobbing, and T. Vardanega. Guide for the use of the ada ravenstar profile in high integrity systems. Technical Report YCS-2003-348, University of York, 2003.
- [6] A. Diller. *LaTeX wiersz po wierszu*. Wydawnictwo Helion, Gliwice, 2000.
- [7] Rafael Gonzalez and Richard Woods. *Digital Image Processing Global Edition*. Pearson Deutschland, 2017.
- [8] L. Lamport. *LaTeX system przygotowywania dokumentów*. Wydawnictwo Ariel, Kraków, 1992.
- [9] J. Peleska, D. Große, A. E. Haxthausen, and R. Drechsler. Automated verification for train control systems. In *Proc. of the 5th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 2004)*, pages 252–265, Braunschweig, Germany, December 2004.

# Spis rysunków

3.1	Diagram ERD projektowanej aplikacji. . . . .	13
3.2	Przykład użycia \subcaption: (a) litera A, (b) litera B. . . . .	14
5.1	Obraz wejściowy i jego histogram. . . . .	23
5.2	Obraz po wyrównaniu i jego histogram. . . . .	24



# Spis tabel

3.1	To jest tytuł tabeli. . . . .	14
3.2	Klasy trajektorii zadań . . . . .	14

## Spis listingów

3.1	Niezoptymalizowany algorytm NWD wersja rekurencyjna . . . . .	16
3.2	Zoptymalizowany algorytm NWD wersja rekurencyjna . . . . .	16
3.3	Niezoptymalizowany algorytm NWD wersja rekurencyjna . . . . .	17
3.4	Przykładowy listing w JAVA . . . . .	17
3.5	Przykładowy listing w C# . . . . .	18
5.1	Wczytanie obrazu i obliczenie histogramu . . . . .	23
5.2	Wyrównanie histogramu . . . . .	23

Załącznik nr 2 do Zarządzenia nr 228/2021 Rektora Uniwersytetu Rzeszowskiego z dnia 1 grudnia 2021 roku w sprawie ustalenia procedury antyplagiatowej w Uniwersytecie Rzeszowskim

## OŚWIADCZENIE STUDENTA O SAMODZIELNOŚCI PRACY

.....Gabriel Łasicki.....

Imię (imiona) i nazwisko studenta

Wydział Nauk Ścisłych i Technicznych

.....Informatyka.....

Nazwa kierunku

.....gl131465.....

Numer albumu

1. Oświadczam, że moja praca projektowa pt.: Sprawozdanie z ćwiczeń laboratoryjnych z przedmiotu Biometryczne Systemy Zabezpieczeń
  - 1) została przygotowana przeze mnie samodzielnie\*,
  - 2) nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
  - 3) nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,
  - 4) nie była podstawą otrzymania oceny z innego przedmiotu na uczelni wyższej ani mnie, ani innej osobie.
2. Jednocześnie wyrażam zgodę/nie wyrażam zgody\*\* na udostępnienie mojej pracy projektowej do celów naukowo-badawczych z poszanowaniem przepisów ustawy o prawie autorskim i prawach pokrewnych.

\_\_\_\_\_  
(miejscowość, data)

\_\_\_\_\_  
(czytelny podpis/y studenta/ów)

\* Uwzględniając merytoryczny wkład prowadzącego przedmiot

\*\* – niepotrzebne skreślić