



# Praca dyplomowa inżynierska

na kierunku Informatyka

Szablon  $\text{\LaTeX}$  do pracy dyplomowej i  
prezentacji

Paweł Niziołek

numer albumu 102488

promotor

prof. dr hab. inż. Szczepan Paszkiel

Opole, 2024

## Szablon $\text{\LaTeX}$ do pracy dyplomowej i prezentacji

### Streszczenie

Praca dyplomowa skupia się na projektowaniu i realizacji szablonu do tworzenia pracy dyplomowej przy użyciu składu tekstu  $\text{\LaTeX}$ . W pracy zostały przedstawione korzyści korzystania z  $\text{\LaTeX}$ a w porównaniu do innych edytorów tekstu skupiając się na takich rzeczach jak automatyczne formatowania, obsługa bibliografii, profesjonalny wygląd i łatwa edycja struktury dokumentu. Teoretyczna część pracy jest poświęcona przedstawieniu krótkiej historii powstawania  $\text{\LaTeX}$ a oraz korzyści płynących z korzystania z  $\text{\LaTeX}$ a przy tworzeniu profesjonalnie wyglądających i automatycznie formatujących się prac.

W praktycznej części pracy przedstawiono stworzony szablon z przedstawieniem jego funkcji oraz możliwością spersonalizowania go do indywidualnych potrzeb użytkownika. Opisano również jego proces instalacji i konfiguracji tak aby ułatwić korzystanie z niego innym użytkownikom.

**Słowa kluczowe:** Skład tekstu  $\text{\LaTeX}$ , Praca Dyplomowa, Narzędzia i środowiska pracy

## Szablon $\text{\LaTeX}$ do pracy dyplomowej i prezentacji

### Abstract

The thesis focuses on the design and implementation of a template for creating a thesis using  $\text{\LaTeX}$ text composition. The thesis presents the advantages of using  $\text{\LaTeX}$ a in abduction to other text editors focusing on such things as automatic formatting, bibliography support, professional appearance and easy editing of the document structure. The theoretical part of the work is devoted to presenting a brief history of the creation of  $\text{\LaTeX}$ a and the benefits of using  $\text{\LaTeX}$ a to create professional-looking and automatically formatting works.

The practical part of the work presents a ready-made template with a demonstration of its functions and the possibility of personalizing it to individual user needs. Its installation and configuration process is also described so as to facilitate its use by other users.

**Keywords:** Text Composition  $\text{\LaTeX}$ , Dissertation, Tools and working environments.

## **Spis treści**

# Rozdział 1

## Wprowadzenie

W prezentacjach profesjonalnych dokumentów ważną rolę odgrywa skład tekstu. W zakresie dostępnych rozwiązań,  $\text{\LaTeX}$  pokazał się jakieś mocne narzędzie pozwalające tworzyć teksty wysokiej jakości, zwłaszcza jeśli chodzi o artykuły książki i prace naukowe. Praca ma na celu stworzenie szablonu  $\text{\LaTeX}$  do pracy dyplomowej i prezentacji. W następnych rozdziałach omówione zostaną informacje teoretyczne dotyczące formatowania dokumentów  $\text{\LaTeX}$ . W rozdziale 8 zostało omówione sposób korzystania z szablonu przygotowanego przez autora pracy.

$\text{\LaTeX}$  jest systemem składu tekstu opartym na  $\text{\TeX}$  stworzonym przez Donalda E. Knutha [3], a sam został napisany przez Lesliego Lamporta w latach 80 [6]. XX wieku.  $\text{\TeX}$  został opracowany jako system składu tekstu przez Donalda Knutha w latach 70. XX wieku, który chciał stworzyć narzędzie umożliwiające precyzyjne i estetyczne składanie tekstu, przede wszystkim w dziedzinie matematyki i nauk komputerowych. W roku 1983 została opublikowana pierwsza wersja  $\text{\LaTeX}$  przez Leslie Lamport, stanowiąca zestaw makr na bazie  $\text{\TeX}$ a. Jednak  $\text{\TeX}$  był bardzo skomplikowanym narzędziem dla osób nieznających go, a  $\text{\LaTeX}$  miał uprościć i ułatwić pracę dokumentami. Zaczął zyskiwać popularność wśród naukowców, inżynierów, matematyków oraz osób potrzebujących profesjonalnego narzędzia do składania tekstu w dziedzinach naukowych.  $\text{\LaTeX}$  był otwartym oprogramowaniem (ang. open-source), co wpłynęło na rozwój społeczności użytkowników. Dzięki szerszej współpracy pojawiły się pakiety, klasy dokumentów oraz szablony, które zwiększyły funkcjonalność samego  $\text{\LaTeX}$ a. W następstwie pojawiły się różne rozszerzenia  $\text{\LaTeX}$ a, takie jak na przykład  $\text{\XeLaTeX}$  czy  $\text{\LuaLaTeX}$ , które umożliwiły nowe możliwości, obsługę formatów czcionek oraz systemów znaków.  $\text{\LaTeX}$  to dynamiczny system składu tekstu, a jego różne dystrybucje, takie jak  $\text{\TeX}$  Live czy  $\text{\MiKTeX}$ , są regularnie aktualizowane i rozwijane. Społeczność użytkowników aktywnie pracuje nad nowymi funkcjami i poprawkami. Szablony  $\text{\LaTeX}$  są stosowane wszędzie tam, gdzie wymagana jest precyzja i wysoka jakość składu tekstu.  $\text{\LaTeX}$  pozostaje popularnym narzędziem składu tekstu, ponieważ jest stabilnym i elastycznym. Stale ewoluuje, dostosowując się do zmieniających się potrzeb użytkowników oraz nowych standardów w składzie tekstu.

Zalety wynikające z korzystania z narzędzia  $\text{\LaTeX}$  są ogromne, dzięki czemu jest szeroko uznanym narzędziem w środowiskach naukowych. Oferuje wysoką jakość składu tekstu, umożliwia

---

elastyczny i profesjonalny wygląd dokumentu. Dla prac naukowych, artykułów i innych dokumentów o charakterze naukowym,  $\text{\LaTeX}$  stanowi kluczowy element. Zaawansowana topologia systemu pozwala na dokładne formatowanie tekstu, tabel czy równań matematycznych. Dzięki złożonej strukturze  $\text{\LaTeX}$  pozwala na tworzenie własnych makr i pakietów, a także skomplikowanych struktur dokumentu takie jak nagłówki, pod nagłówki, sekcje, podsekcje, gdzie autorzy mają kontrolę nad formatowaniem każdego poziomu struktury. Posiada również fachowe narzędzie do zarządzania cytowaniami i bibliografiami, gdzie system automatycznie generuje bibliografię zgodnie z wybranym stylem. Umożliwia wydzielenie zawartości od formy znaczy to, że autorzy mogą się skupić na samym tekście podczas gdy system sam zajmie się formatowaniem. Zapewnia stabilną i niezawodną pracę na różnych systemach operacyjnych i jest dostępny w formie otwartego oprogramowania (ang. open-source), co pozwala na rozwój i wsparcie od strony społeczności  $\text{\LaTeX}$ . Dzięki rozwojowi technologii jest także możliwość korzystania z systemu  $\text{\LaTeX}$  poprzez internetowe edytory takie jak na przykład Overleaf. Automatyczna numeracja spisu treści czy indeksów umożliwia pracę z dużymi dokumentami taki jak na przykład książki, rozprawy doktorskie. Autorzy nie muszą się martwić detalami formatowania, ponieważ  $\text{\LaTeX}$  zajmuje się nimi.  $\text{\LaTeX}$  ułatwia pracę w zespole nad jednym dokumentem, co jest często wykorzystywane w badaniach naukowych. Zmiany są dokonywane w czytelny sposób, dlatego ułatwia to śledzenie i zrozumiałość. Narzędzie wspiera wiele języków i umożliwia korzystanie z różnych znaków specjalnych czy akcentów, co pozwala na redagowanie tekstów które używają znaków diakrytycznych takich jak (ą, ę, ć). Obsługuje również różne standardy kodowań znaków takich jak na przykład UTF-8 Latin-1 czy CP1250. Te wyżej wymienione zalety sprawiają, że  $\text{\LaTeX}$  jest idealnym narzędziem dla osób wymagających precyzji i profesjonalizmu w swoich pracach.

Pomimo tego że  $\text{\LaTeX}$  ma wiele zalet, istnieją również pewne wady, które mogą sprawić, że nie będzie odpowiedni dla wszystkich użytkowników.  $\text{\LaTeX}$  ma duży próg wejścia dla nowych użytkowników, osoby które nie miały wcześniej kontaktu z  $\text{\LaTeX}$  będą potrzebowały czasu aby nauczyć się podstawowych poleceń albo struktury dokumentu. Obsługiwany jest często poprzez pisanie kodu źródłowego w edytorze. Dlatego osoby przyzwyczajone do pracy z programami oferującymi graficzny interfejs użytkownika, taką formę pracy mogą uznać za mniej intuicyjną. Dla krótkich dokumentów tekstowych takich jak listy czy notatki,  $\text{\LaTeX}$  może być uważany za czasochłonny i zbyt rozbudowany w porównaniu do innych narzędzi do edycji tekstu.  $\text{\LaTeX}$  jest zależny od zewnętrznych narzędzi ponieważ wymaga kompilacji co oznacza że użytkownik musi korzystać z zewnętrznych narzędzi do przetwarzania kodu na gotowy dokument PDF lub inny format. Warto jednak zauważyć że te wady mogą być znoszone poprzez doświadczenie użytkownika a niektóre mogą być łagodzone poprzez korzystanie z odpowiednich edytorów ułatwiających pracę. Podsumowując jest doskonałym narzędziem do tworzenia skomplikowanych dokumentów technicznych, naukowych czy akademickich. Podczas gdy tradycyjne edytory takie jak Microsoft Word czy Google Docs są bardziej dostosowane do prostszych zastosowań i oferują łatwiejszą obsługę dla osób bez fachowej wiedzy.

## Rozdział 2

# Cel i zakres pracy

Celem niniejszej pracy jest stworzenie szablonu do prezentacji i pracy dyplomowej z wykorzystaniem środowiska  $\text{\LaTeX}$ . Głównymi celami podczas tworzenia szablonów są usprawnienie formatowania i estetyki. W kontekście  $\text{\LaTeX}$  dotyczy gwarancji, że szablon spełnia reguły formatowania pracy naukowej również jest czytelny profesjonalny i estetyczny. Aspekty które należy uwzględnić, aby usprawnić formatowanie i estetykę to:

- marginesy i układ strony,
- czcionka i interlinia,
- nagłówki i stopki,
- numeracja rozdziałów i sekcji,
- rozmieszczenie grafik i tabel,
- kolory i formatowanie,
- obsługa wzorów matematycznych,
- dostosowanie do polskich standardów.

Szerzej zostanie to omówione w późniejszych rozdziałach. Zastosowanie wymienionych elementów umożliwi spełnienie wymogów narzuconych ogólnie przez uczelnię, ale także przedstawia się czytelnie i profesjonalnie. Całość powinna być spójna i zgodna z specyfiką pracy dyplomowej.

Zautomatyzowanie struktury dokumentu w  $\text{\LaTeX}$  powinno być tak skonfigurowane, aby umożliwić autorowi skupieniu się na treści bez potrzeby ręcznego dostosowywania struktury. Dlatego wiele aspektów powinno być zautomatyzowane. Elementy takie jak rozdziały, sekcje, podsekcje, spis treści, biografia, numeracja stron, rozmieszczenie rysunków i tabel i nagłówki oraz style stron. Dzięki zmechanizowaniu wymienionych elementów autor może skoncentrować się na treści, a struktura dokumentu zostanie utrzymana w spójny sposób. Zautomatyzowanie układu dokumentów w  $\text{\LaTeX}$ u jest jednym z kluczowych aspektów, z powodu którego wielu studentów i naukowców wybiera ten system celem składania tekstu różnego typu prac.

Zarządzanie bibliografią i przypisami jest kolejnym z kluczowych elementów w tworzeniu pracy.  $\text{\LaTeX}$  dostarcza narzędzia takie jak BibTex czy biber które wspomagają organizację cytowań. Dzięki narzędziom  $\text{\LaTeX}$  zarządzanie bibliografią i przypisami staje się mniej skomplikowane i

---

efektywniejsze. Trzeba zwrócić uwagę, że aby  $\text{\LaTeX}$  poprawnie odnosił się do cytowania i tworzenia spisu bibliograficznego należy po każdym dodaniu pozycji ponownie skompilować dokument.

Dostosowanie do wymagań uczelni polega na spełnieniu określanych standardów, w tym przypadku wymogów Politechniki Opolskiej dotyczących formatowania i struktury prac dyplomowych. Obejmuje to przede wszystkim logo politechniki, układ stron, styl bibliografii czy inne aspekty pracy.

Wspieranie różnych języków pracy szablon umożliwia pisanie pracy zarówno w języku polskim jak i w języku angielskim. Wymaga to przede wszystkim dostosowania szablonu do specyfiki obu języków. Dostosowania takich elementów jak strona tytułowa, cytowania, numeracja rozdziałów itp. wymaga zastosowania odpowiednich komend, a także spójności i precyzji w uwzględnianiu zasad gramatyki czy stylistyki.

Przyjazność dla użytkownika polega to na dostosowaniu szablonu tak żeby nie był zbyt skomplikowany pod względem pracy z nim. Powinien ułatwiać pracę szczególnie tym użytkownikom, którzy nie mieli wcześniej styczności z  $\text{\LaTeX}$ em. W procesie tworzenia przyjaznego szablonu w  $\text{\LaTeX}$ , takie elementy jak czytelność kodu, dokumentacja i intuicyjna modyfikacja są kluczowe. Jest to jedno z głównych założeń pracy. Zakres pracy obejmuje realizację wyżej wymienione punkty a ponadto szablonu z udziałem użytkownika celem wyłapania niedociągnięć czy uwzględnienia sugestii.

## Rozdział 3

# Opis technologii

Struktura dokumentów  $\LaTeX$  przedstawiona w listingu ?? obejmuje proste i podstawowe polecenia. Dokumenty  $\LaTeX$  zaczynamy od **preambuły** która znajduje się przed właściwą treścią dokumentu. Preambuła zawiera instrukcje i ustawienia, które wpływają na cały dokument. Najczęściej zawiera takie informacje jak: klasa dokumentu, ustawienia strony czy pakiety. W przypadku podanego przykładu zaczynamy od wyboru klasy którym jest `article`, więcej informacji na temat wyboru klas w dokumentach  $\LaTeX$  można znaleźć w podrozdziale ?. Następnym elementem są pakiety w  $\LaTeX$  to zestawy makr i definicji, które rozszerzają funkcjonalność języka  $\LaTeX$ . Dzięki pakietom możesz dostosować formatowanie, dodawać nowe polecenia czy obsługiwać specjalne elementy, takie jak tabelki, grafika czy dodatkowe czcionki. Wczytujemy pakietu w  $\LaTeX$  za pomocą polecenie `\usepackage{Nazwa_pakietu}` jednak należy pamiętać że niektóre pakiety mogą wchodzić z sobą w konflikt. Kolejnym elementem przedstawionym w listingu ?? jest makro to niestandardowe polecenia, które można definiować i używać w celu skrócenia i uproszczenia kodu. Makra pozwalają na zdefiniowanie własnych komend, które mogą być używane w dokumencie. Mogą być one przydatne, gdy chcemy wielokrotnie używać tego samego fragmentu kodu, lub gdy chcemy nadać bardziej znaczącą nazwę bardziej skomplikowanym lub długim sekwencjom poleceń. W przedstawionym przykładzie zostało stworzone makro wstawiającą hiperłącze do platformy Overleaf. Kolejnym elementem są metadane dokumentu to informacje charakteryzujące sam dokument, takie jak tytuł, autor, data itp. Ostatnim elementem w przedstawionym przykładzie jest właściwa treść dokumentu która nie jest już częścią preambuły zaczyna się od `\begin{document}` a kończy na `\end{document}` wszystkie elementy zawarte pomiędzy tworzą treść. Element `\maketitle` generuje stronę tytułową na podstawie zdefiniowanych metadanych. Za pomocą komendy `\section{Nazwa_sekcji}` tworzona jest nową sekcje w naszym dokumencie.



---

**Listing 1.** Przykład prostego dokumentu

```
1 \documentclass{article}
2
3 \usepackage[utf8]{inputenc} % Kodowanie znak w
4 \usepackage[T1]{fontenc} % Kodowanie fontu
5 \usepackage[polish]{babel} % J ęzyk dokumentu
6 \usepackage{graphicx} % Obs ługa grafiki
7 \usepackage{hyperref} % Dodanie pakietu do obs ługi hiper łączy
8
9 % Makro
10 \newcommand{\overleafflink}{\href{https://www.overleaf.com/}{Overleaf }}
11
12 \title{Przykładowy Dokument w LaTeX}
13 \author{Wojtek Hyl}
14 \date{\today}
15
16 \begin{document}
17
18 \maketitle
19
20 \section{Wprowadzenie}
21 Tutaj mo esz opisa wprowadzenie do swojego dokumentu.
22
23 \section{Link do Overleaf}
24 Korzystaj c z makra, mo esz wstawi hiper łączy do Overleaf: \
25 overleafflink
26 \end{document}
```

---

Overleaf to platforma internetowa umożliwiająca pracę z dokumentami  $\text{\LaTeX}$ , która ułatwia autorom pisanie i formatowanie tekstów naukowych. Platforma oferuje również dostęp do szablonów udostępnionych przez użytkowników. Dzięki temu, że jest to platforma online nie ma potrzeby instalowania żadnych dodatkowych programów na lokalnym komputerze. Ułatwia to dostępność i pracę z dowolnego miejsca z dostępem do internetu. Możliwość pracy online umożliwia pracę w zespołach nad jednym dokumentem w czasie rzeczywistym co jest bardzo ważnym elementem. Overleaf umożliwia integrację z platformą GitHub, co umożliwia przechowywanie dokumentów  $\text{\LaTeX}$  w systemie kontroli wersji. Ułatwia to przechowywanie wersji projektu, śledzenie zmian oraz współpracę z innymi programistami. Overleaf oferuje również integrację z Dropboxem która umożliwia przechowywanie i synchronizację plików w chmurze. Dropbox jest bardziej intuicyjnym narzędziem do przechowywania kodów niż GitHub jednakże nie posiada kontroli wersji. Platforma oferuje bogaty zasób funkcji takich jak na przykład natychmiastowy podgląd, sprawdzanie poprawności kodu, sugestie poprawek i wiele innych przydatnych funkcjonalności. Dodatkowo strona obsługuje wiele języków, co umożliwia pisanie oraz formatowanie w wybranym języku. Bibliografia na platformie jest

---

zintegrowana z systemem zarządzania bibliografią. Pomaga to w generowaniu bibliografii z wybranym stylem. Posiada integracje z takim portalami jak Mendeley i Zotero, co umożliwia importowanie i eksportowanie bibliografii w różnych formatach, a także pozwala na integrację z zarządzaniem referencjami. Overleaf oferuje także bardzo bogatą i łatwo dostępną dokumentację oraz instrukcję obsługi pakietów. Podsumowując platforma ułatwia tworzenie dokumentów  $\text{\LaTeX}$  dla osób, które nie chcą instalować żadnego dodatkowego oprogramowania swoim komputerze lokalnym i chcą mieć dostęp do swoich dokumentów w każdym miejscu. Jest również świetnym narzędziem do pracy w zespołach które pracują online nad swoimi dokumentami. Należy jednak pamiętać że takie funkcje jak integracja z GitHubem, DropBoxem, Mendeley i Zotero są wyłącznie dostępne w płatnej subskrypcji Overleaf. Płatna wersja również zawiera wsparcie priorytetowe (ang. priority support) co umożliwia szybsze wsparcie z strony Overleaf w razie problemów.

$\text{\LaTeX}$  umożliwia również pracę w środowisku lokalnym. Istnieje wiele narzędzi używanych do tworzenia, kompilowania i zarządzania dokumentami  $\text{\LaTeX}$ . Jedną z istotniejszych narzędzi jakie są potrzebne to edytory. Do najczęściej używanych zaliczamy TeXShop, TeXworks i TeXstudio. Edytor TeXShop umożliwiający pracę z  $\text{\LaTeX}$ em na systemy OS X. Posiada prosty interfejs i jest łatwy w użyciu. Natomiast jeżeli chodzi o edytory dostępne na wielu systemach możemy do nich zaliczyć TeXworks i TeXstudio. Kolejną ważną rzeczą jaka jest potrzebna do używania  $\text{\LaTeX}$ a na komputerach lokalnych jest jego dystrybucja. Do najczęściej używanych dystrybucji należą TeX Live i MiKTeX. TeX Live jest to dystrybucja  $\text{\LaTeX}$  dostępna na wielu platformach obejmująca pełen zestaw pakietów i narzędzi. MikTeX jest dystrybucją  $\text{\LaTeX}$  przeznaczoną dla systemów Windows podobnie jak TeX Live zawiera pełen zestaw pakietów i narzędzi. Ostatnią rzeczą jaka jest potrzebna do pracy z  $\text{\LaTeX}$ em jest kompilator. pdf $\text{\LaTeX}$  jest najczęściej używanym kompilatorem umożliwiającym bezpośrednio generację plików PDF. Obsługuje nowoczesne funkcje, takie jak przykład na hiperłącza, wbudowane obrazy oraz obsługę różnych fontów. Jednym z kolejnych kompilatorów jest Xe $\text{\LaTeX}$ . Obsługuje on fonty systemowe co umożliwia użycie innych fontów niż standardowe w  $\text{\LaTeX}$ . Obsługuje również Unicode, co pozwala na pracę w wielu językach i kodowań znaków. Lua $\text{\LaTeX}$  jest edytorem umożliwiające zaawansowane operacje programistyczne używające silnika LuaTeX jest rozwinięciem pdf $\text{\LaTeX}$ .

## Rozdział 4

# Usprawnienie formatowania i estetyki

### 4.1 Marginesy i układ strony

W szablonach  $\text{\LaTeX}$  ustawienia dotyczące marginesów i układu strony są kluczowe, ponieważ wpływają na końcowy wygląd dokumentu. Jednym z ważniejszych rzeczy jest odpowiedni wybór klasy dokumentu, która jest zgodna z wymaganiami. Każda klasa ma swoje specyficzne cechy i ustawienia, które można edytować do indywidualnych potrzeb. Do najczęściej używanych klas należą `article`, `report` czy `letter`. O właściwościach klas można się dowiedzieć czytając informacje na ich temat w rozdziale 8 książki [2].  $\text{\LaTeX}$  umożliwia ustawianie układu strony do własnych potrzeb. Ustawianie marginesów jest ważnym elementem w kontekście estetyki. Do ustawiania marginesów w  $\text{\LaTeX}$  służy pakiet `geometry`, który umożliwia ich dokładnie ustawianie. Marginesy w ogólnym przypadku określają pustą przestrzeń między kolumną tekstu a krawędziami strony. Właściwie dostosowane wpływają na estetykę i ogólny odbiór dokumentu. W większości przypadków, np. w pracy dyplomowej ich wartość ustawia się zazwyczaj na 2,5 cm. Jeśli chodzi o  $\text{\LaTeX}$  wartości marginesów można ustawić za pomocą wspomnianego wyżej przedstawionego pakietu `geometry`. Domyślne ustawienia zależą jednak od wybranej klasy dokumentu, jednak bardzo łatwo można dostosować ich wartość do własnych potrzeb.

Kolejnym elementem w kontekście formatowania i estetyki jest określenie czy dokument ma być jednostronny czy dwustronny. Ważne jest to o tyle, że dla dokumentów dwustronnych mogą występować różnice, jeśli chodzi o numerację stron i marginesy, jeśli chodzi o strony parzyste czy nieparzyste. W układzie dwustronnym, dostosowanie numeracji stron jest istotne, aby zachować spójność i estetykę wizualną. Często stosuje się różne układy numeracji stron na stronach lewych i prawych, a także dostosowuje ich położenie, na przykład umieszczając numerację stron na środku strony. Określenie pionowej i poziomej justyfikacji wspomaga pakiet `ragged2e`. Pozwala on na elastyczniejszą justyfikację tekstu w obrębie tych kierunków.  $\text{\LaTeX}$  pozwala również na określenie innych parametrów takie jak rozmiar papieru, odstęp między tekstem a nagłówkiem czy też między stopką a tekstem. Ważne jest, aby dostosować te wartości do własnych preferencji czy też określonych wymagań. Odpowiednie przystosowanie tych elementów pozwala na stworzenie

dokumentu o elastycznym układzie. Gwarantuje to również profesjonalny wygląd a sam dokument staje się bardziej przyjemny w odbiorze.

## 4.2 Czcionka i interlinia

Elementy takie jak czcionka i interlinia wpływają drastycznie na wygląd dokumentu, ponieważ źle wybrany font tekstu czy odstęp pomiędzy wierszami utrudniają czytelność i pogarszają wygląd dokumentu. Czcionkę dostosowujemy przy użyciu pakietu `fontspace`. Jednak trzeba pamiętać, że nie wszystkie fonty są dostępne w domyślnej wersji systemu, więc używanie niestandardowych wymaga ich dodania za pomocą pakietu `fontspec`. Interlinię w  $\text{\LaTeX}$  możemy dostosowywać za pomocą pakietu takiego jak `setspace`. Ustawienie odstępu między wierszami na 1 będzie skutkowało tym, że odstęp będzie równy wysokości czcionki. Analogicznie ustawienie go na 2 będzie zwiększało go dwukrotnie. W stworzony przez autora szablonie została użyta czcionka `Latin Modern Roman` która jest główną krojem pisma. Natomiast czcionka `Latin Modern Math` została zastosowana do wzorów matematycznych.

## 4.3 Nagłówki i stopki

Nagłówki i stopki w  $\text{\LaTeX}$  są obsługiwane przez na przykład pakiet `fancyhdr`. Pakiet umożliwia regulację górnych (nagłówki) i dolnych (stopki) elementów każdej strony. Oferuje on możliwość dostosowywania takich elementów jak na przykład linie oddzielające od treści, odległości, numerację stron i umieszczanie nazw rozdziałów. Umożliwia także ustawianie różnych wyglądków stopki i nagłówka dla stron parzystych i nieparzystych. Ważne jest, aby dostosowywać te elementy do potrzeb własnej pracy.

## 4.4 Numeracja rozdziałów i sekcji

Numeracja rozdziałów i sekcji jest automatycznie prowadzona, jednakże jest możliwość dostosowania w jaki sposób te numery są formatowane czy które struktury mają być numerowane. Domyślnie każdy rozdział jest przedstawiony w formie numeru rozdziału przed jego tytułem.  $\text{\LaTeX}$  za pomocą pakietu `titlesec` umożliwia zmienianie formy wyświetlania. Sekcje podobnie jak rozdziały są numerowane. Automatycznie możemy kontrolować, które sekcje są numerowa za pomocą komendy `setcounter`, jednak należy pamiętać, że nienumerowane sekcje nie pojawią się w spisie treści. Pakiet `titlesec` również umożliwia kontrolę i edycję sekcji. Warto mieć na uwadze, że zbyt duża ilość numeracji poziomów może zmniejszać czytelność dokumentu i profesjonalny wygląd.

## 4.5 Rozmieszczenie grafik i tabel

W wielu dokumentach grafiki są istotnym elementem przedstawienia treści. Pakiet `graphicx` jest narzędzie dodające możliwość wstawiania grafik w  $\text{\LaTeX}$ . Otoczenie `figure` pozwala na umieszczanie grafiki, a także umożliwia również kontrole i numerację wstawionej treści w dokumencie. Umieszczone w ten sposób grafiki automatycznie są podpisywane i opisywane, a sam spis grafik jest generowany w sposób zautomatyzowany. Kolejnym istotnym elementem przedstawiania treści w  $\text{\LaTeX}$  są tabele. Dzięki otoczeniu `table` istnieje precyzyjna kontrola ich położenia oraz numeracja, podobnie jak w środowisku `figure`. Spis tabel jest generowany automatycznie i może być wstawiony w dowolnym miejscu w dokumencie. Istnieje wiele edytorów internetowych które ułatwiają tworzenie tabel dla osób, które nie są zaznajomione z  $\text{\LaTeX}$ em lub preferują wizualne interfejsy do projektowania tabel. Pozwalają one generować kod  $\text{\LaTeX}$  bez konieczności ręcznego wpisywania skomplikowanych komend. Rozmieszczenie grafik i tabel wymaga dobrego poznania struktury i użycia odpowiednich środowisk. Więcej na temat użycia otoczenia `figure` w podrozdziale ?? i na temat otoczenia `table` w podrozdziale ?. Informacji na temat mechanizmu umieszczania obiektów ruchomych można znaleźć np. w podrozdziale 2.11 publikacji [9] lub podrozdziale 8.4.1 książki [5]. Ważnym elementem jest zadbanie o odpowiednie podpisy co znacząco ułatwia nawigację po dokumencie.

## 4.6 Kolory i formatowanie

$\text{\LaTeX}$  jest narzędziem umożliwiającym dostosowywanie koloru tekstu do potrzeb dokumentu. Pakietem umożliwiającym kontrolę nad kolorem jest `xcolor`. Dzięki temu pakietowi można dostosować kolor tekstu i tła. Obsługuje kolory podawane po nazwie, RGB, HTML, CMYK i szaro-skali. Stosowanie kolorów w dokumentach czy prezentacjach daje możliwość podkreślenia ważnych elementów, poprawia czytelność i zrozumiałość, a w prezentacjach poprawia atrakcyjność wizualną pracy. Jednakże pamiętać należy, że zbyt intensywne czy chaotyczne kolory przeszkadzają i utrudniają odczyt treści.  $\text{\LaTeX}$  jak większość edytorów tekstowych daje możliwość formatowania tekstu takich jak pogrubienia, pochylenia czy podkreślenia.

## 4.7 Obsługa wzorów matematycznych

$\text{\LaTeX}$  to idealne narzędzie do obsługi wzorów matematycznych. Umożliwia profesjonalne i precyzyjne przedstawianie matematyki. Dwoma głównymi trybami obsługi matematyki są tryb wewnętrzny (`inline`) i tryb wyłączony (`display`). In-line umożliwia osadzenie wzorów w tekście, natomiast tryb `display` umieszcza wzór w osobnej linii dokumentu. Symbole matematyczne są zapisywane w kodzie za pomocą znaków takich jak `+`, `-`, `*`, `/`. Istnieje również możliwość potęgowania za pomocą znaku <sup>^</sup> i pierwiastkowania za pomocą funkcji `sqr`t.  $\text{\LaTeX}$  również obsługuje funkcje i operacje matematyczne takie jak na przykład sinus, cosinus, logarytm czy suma. Obsługuje również symbole takie jak `in` (należy do) czy `neq` (różne od). Pozwala na tworzenie macierzy

za pomocą środowiska `bmatrix` (macierze kwadratowe) czy `pmatrix` (macierze okrągłe). W sieci dostępne są również edytory które pomagają w tworzeniu wzorów matematycznych poprzez wizualizację podanego przez nas wzoru i przetworzenie go w komendy  $\text{\LaTeX}$ . Kompletne zestawienie symboli, dostępnych standardowo w trybie matematycznym, można znaleźć w podrozdziale ?? lub w podrozdziale 3.9 publikacji [9] lub w dodatku A książki [2].

## Rozdział 5

# Automatyzacja struktury dokumentu

Automatyzację struktury dokumentu można uzyskać używając odpowiednie środowiska i polecenia. Pozwalają one na sprawne i dynamiczne formatowanie różnych części takich jak rozdziały, podrozdziały itp. Jedną z technik, która wspomaga automatyzację dokumentu  $\text{\LaTeX}$  są polecenia strukturalne. W  $\text{\LaTeX}$ u, aby automatycznie wygenerować numerowanie i spis treści dla sekcji czy też podrozdziałów itp. należy użyć odpowiednich poleceń. Między innymi są to polecenia takie jak `\section{sekcja}`, `\subsection{Podsekcja}` itd. Kolejnym elementem, który wspomaga automatyzację dokumentu jest zastosowanie pętli. System składu tekstu  $\text{\LaTeX}$  oferuje pętle takie `for`, `foreach` z pakietu `pgffor`. Pozwalają na automatyczne powtarzanie określonych operacji, co jest szczególnie użyteczne, gdy chcemy zastosować tę samą strukturę do różnych danych czy warunków. Numeracja ustawiana jest automatycznie, jeśli chodzi o sekcje, równania czy rozdziały. Jednak, aby uniknąć ręcznego ustawienia numeracji warto zastosować automatyczne etykiety i odwołania. Realizowane jest to przez użycie takich poleceń jak `\label{sec:sekcja1}` czy `\ref{sec:sekcja}`. W  $\text{\LaTeX}$ u można definiować makra, co umożliwia dynamiczne generowanie wielokrotnie używanych struktur. Makra w  $\text{\LaTeX}$  to rodzaj zdefiniowanych komend, które pozwalają na zgrupowanie określonych operacji lub tekstu pod jednym identyfikatorem. Ważnym jest również wybór konkretnej klasy dokumentu w zależności od potrzeb czy ogólnie narzuconych wymagań. Wśród sekcji klasa dokumentu można wybrać struktury takie jak np. `report` lub `book`. Klasa `report` oferuje rozdziały a `book` także części. Pakiet `titlesec` pozwala dostosować formatowanie nagłówek sekcji, a co za tym idzie jest pomocna w pewnych elementach automatyzacji takiego dokumentu. Jeśli chodzi o szablon można utworzyć go z zdefiniowanymi elementami. Szablon może być udostępniany innym użytkownikom jak również wykorzystany w innych projektach. Elementy opisane w tej części znacząco wspomagają i poprawiają automatyzację dokumentu. Pozwalają utrzymać spójność a także poprawiają pracę nad strukturą dokumentów, szczególnie gdy są one bardzo rozbudowane.

## Rozdział 6

# Zarządzanie bibliografią i przypisami

Jest to następny element a także kluczowy aspekt w kontekście tworzenia prac naukowych czy dokumentów akademickich. W tym celu również używane są specjalne pakiety między innymi wspomniany wcześniej BibTeX, a także różne komendy służące do dodawania przypisów w tekście. Narzędzie to wspomaga zarządzanie bibliografią. Gromadzi dane o cytowanych pracach z różnych źródeł. Automatycznie formatuje je zgodnie z wybranym stylem. Pakiet dołączany jest do dokumentu poprzez komendę `\usepackage{natbib}`. W celu przechowywania informacji o źródłach, które są cytowane w dokumencie należy utworzyć plik `nazwa_bibliografi.bib`. Przykładowa struktura takiego pliku:

**Listing 2.** Przykłady dodawania do bibliografii

```
1 %Artykul
2 @article{id,
3   author = {Autor, A.},
4   title = {Tytul artykulu},
5   journal = {Czasopismo},
6   year = {2000},
7   pages = {123-145},
8 }
9 %Ksiazka
10 @book{id,
11   author = {Imie Nazwisko},
12   title = {Tytul ksiazki},
13   publisher = {Wydawnictwo},
14   year = {2005},
15   address = {Miasto},
16   edition = {2},
17 }
18 %Strona internetowa
19 @online{id,
20   author = {Imie Nazwisko},
21   title = {Tytul strony internetowej},
```



---

```
22 | year = {2021},  
23 | url = {http://www.example.com},  
24 | }
```

---

Elementy umieszczone w pliku `nazwabibliografi.bib` można zacytować w dokumencie za pomocą polecenia `\cite{id}`. Należy również wybrać styl bibliografii, który definiuje jak formatowane są cytowania. Zazwyczaj odbywa się to za pomocą komendy `\bibliographystyle`. Określenie miejsca, gdzie ma zostać umieszczona bibliografia realizowane jest przez dodanie komendy `\bibliography`. Zarządzanie przypisami umożliwia pakiet `footnote`. Pozwala on na łatwe dodanie przypisu dolnego.  $\text{\LaTeX}$  pozwala na dodanie przypisów również w obszarze wzorów matematycznych. W celu oznaczenia przypisu należy użyć `footnotemark`. Dodanie tekstu do przypisu umożliwia z kolei komenda `\footnotetext{treść}`. Przypisy mogą również być sformatowane zgodnie z odpowiednim stylem. W tym celu należy użyć pakietu `footmisc`. Zarządzanie bibliografią i przypisami wspomaga efektywne tworzenie dokumentów. Korzystanie z pakietu `BibTeX` znacząco ułatwia cytowanie prac i zarządzanie przypisami. Więcej informacji na temat tworzenia bibliografii można znaleźć na internetowej platformie edukacyjnej [12] lub w artykule [1].

## Rozdział 7

# Wspieranie pracy w języku polskim

Obejmuje kilka elementów, które należy mieć na uwadze realizując ten aspekt. Wpływa to przede wszystkim na poprawne formatowanie i obsługę polskich znaków diakrytycznych. Znakami diakrytycznymi są takie litery jak ą, ę, ó, ć itp. Jeśli chodzi o kodowanie znaków należy się upewnić, że dokument jest zapisywany w kodowaniu UTF-8. Zastosowanie takiego kodowania umożliwia ich poprawną obsługę. Realizowane jest to za pomocą pakietu `\usepackage[utf8]{inputenc}`. W celu skorzystania z polskich liter należy włączyć pakiet `polski` lub `babel` z ustawieniem języka na polski. Jeśli w dokumencie używane są komendy generujące datę, to w takim przypadku trzeba dostosować ją do polskiego formatu. Zrealizować to można pakietem `\usepackage{datetime}` `\renewcommand{dateseparator}{.}`. W zależności od klasy dokumentu, w szczególnym przypadku `report` lub `article`, tytuły takie jak rozdział czy sekcja mogą zostać dostosowane do języka polskiego np.

**Listing 3.** Dostosowywanie nazwa do języka polskiego

```
1 \setglossarysection{section}
2 \makeglossaries
3 \loadglsentries{Glossary}
4 \ifnum\strcmp{\locallang}{PL} = 0
5 \newcommand{\acronymstitle}{Wykaz skr t w i symboli}
6 \else
7 \newcommand{\acronymstitle}{List of Abbreviations and Symbols}
8 \fi
```

Jeśli w dokumencie użyte zostały przypisy należy dostosować je do konwencji polskiej. Także bibliografia powinna być dostosowana do polskich wymagań. Należy upewnić się również czy pozostałe elementy jak stopka, nagłówek czy strona tytułowa są zgodne z polskimi standardami. Wspieranie pracy w języku polskim oferuje konfigurację różnych aspektów. Przede wszystkim należy dostosować je do polskich standardów dla dokumentów akademickich czy naukowych. Odpowiednie ustawienia i dostosowania pozwalają stworzyć spójny a także profesjonalnie wyglądający dokument w języku polskim.

## Rozdział 8

# Projekt szablonu pracy dyplomowej

Podczas tworzenia szablonu do pracy dyplomowej została stworzona klasa dokumentu  $\text{\LaTeX}$  o nazwie `Dyplom.cls`. Zaprojektowana do tworzenia pracy dyplomowej, umożliwiającą wybór różnych opcji, takich jak typ pracy (inżynierska, magisterska, licencjacka), język dokumentu (polski, angielski), a także zawiera szereg ustawień dotyczących formatowania tekstu, czcionek, kolorów i innych elementów. Ponadto, dostarcza funkcje takie jak generowanie wykazu skrótów i symboli, konfiguracja języka w zależności od silnika kompilacji oraz ustawienia dotyczące obrazów, tabel i dodatków. W stworzonej klasie zostały użyte następujące pakiety:

- **indentfirst**: Włącza wcięcie pierwszego akapitu w dokumencie.
- **iftex**: Pozwala na warunkowe wykonywanie kodu w zależności od używanego silnika  $\text{\LaTeX}$ .
- **pdfscape**: Umożliwia obracanie stron w układzie poziomym w dokumentach PDF.
- **amssfonts**: Zapewnia dostęp do dodatkowych fontów matematycznych dostarczanych przez American Mathematical Society (AMS).
- **amsmath**: Rozszerza możliwości składu matematycznego w  $\text{\LaTeX}$ .
- **amssymb**: Zapewnia dostęp do dodatkowych symboli matematycznych dostarczanych przez AMS.
- **amsthm**: Umożliwia definiowanie środowisk teorematycznych i dowodowych.
- **esint**: Dostarcza symboli całkowych w stylu Euler Script.
- **ifXeTeX**: Warunkowe wykonanie kodu w zależności od używanego silnika  $\text{\LaTeX}$ .
- **mathspec**: Umożliwia dostosowanie fontów matematycznych w  $\text{\LaTeX}$ .
- **polyglossia** lub **babel** (w zależności od używanego silnika): Pakiety do obsługi wielojęzyczności w  $\text{\LaTeX}$ .
- **tocbibind**: Kontroluje zawartość spisu treści, listy ilustracji itp.
- **titletoc**: Pozwala na dostosowanie wyglądu spisu treści.
- **xcolor**: Umożliwia obsługę kolorów w dokumentach  $\text{\LaTeX}$ .
- **circuitikz**: Pakiet do rysowania obwodów elektrycznych w  $\text{\LaTeX}$  przy użyciu pakietu TikZ.
- **pgfplots**: Narzędzie do tworzenia wykresów w  $\text{\LaTeX}$  przy użyciu pakietu TikZ.
- **float**: Umożliwia kontrolę położenia floatów (np. obrazków i tabel) w dokumencie.

- 
- **graphicx**: Obsługuje wstawianie grafik (obrazków) do dokumentów  $\LaTeX$ .
  - **etoolbox**: Zapewnia narzędzia do manipulacji listami i warunkami logicznymi w  $\LaTeX$ .
  - **enumitem**: Rozszerza możliwości konfiguracji list w  $\LaTeX$ .
  - **tikz**: Narzędzie do rysowania grafiki wektorowej w  $\LaTeX$ .
  - **glossaries**: Umożliwia zarządzanie indeksem pojęć (glosariuszem) w dokumentach  $\LaTeX$ .

Został stworzony również plik `Dyplom.sty` który stanowi rozszerzenie dla1 klasy dokumentu. Zawiera konfiguracje związane z bibliografią, formatem strony, nagłówkami i stopkami, a także ustawienia dotyczące akapitów, tabel, listingów programów i ozdóbników ścieżek. Ponadto, definiuje makra ułatwiające korzystanie z często używanych elementów, takich jak linki do Overleaf czy wyróżniony tekst maszynowy. Podczas tworzenia rozszerzenia do klasy `Dyplom` zostały użyte takie pakiety jak:

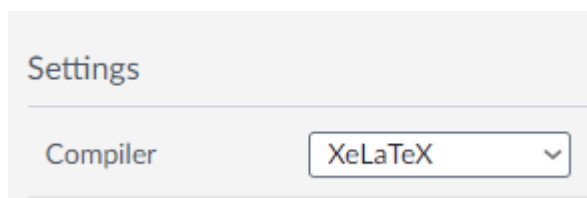
- **kvoptions**: Obsługa opcji konfiguracyjnych w pakietach LaTeX.
- **bibLaTeX**: Zaawansowany system zarządzania bibliografią.
- **chngcntr**: Kontrola numeracji w sekcjach, rysunkach itp.
- **geometry**: Umożliwia konfigurację układu strony.
- **setspace**: Zapewnia łatwą kontrolę interlinii.
- **fancyhdr**: Pozwala na dostosowywanie nagłówków i stopoków w dokumencie.
- **emptypage**: Automatycznie wstawia puste strony na końcu rozdziałów.
- **caption**: Umożliwia dostosowanie podpisów pod rysunkami i tabelami.
- **subcaption**: Dodaje wsparcie dla podpisów dla podobrazków.
- **multirow**: Umożliwia łączenie komórek w tabelach w pionie.
- **multicol**: Pozwala na tworzenie wielokolumnowych bloków tekstu.
- **longtable**: Obsługuje długie tabele rozciągające się przez kilka stron.
- **colortbl**: Dodaje kolorowanie do tabel.
- **listings**: Wstawia kod źródłowy z odpowiednią kolorystyką i numeracją linii.
- **menukeys**: Pozwala na dodawanie klawiszy i skrótów klawiaturowych do dokumentów.
- **hyperref**: Tworzy hiperłącza w dokumencie PDF.
- **csquotes**: Obsługuje cytaty i cudzysłowy w zależności od języka.
- **lipsum**: Generuje losowy tekst (lorem ipsum) do testowania układu.
- **bredzenie**: Pakiet do generowania losowych treści.

Klasa `Dyplom.cls` oraz rozszerzenie `Dyplom.sty` są kompleksowe i mają na celu ułatwienie tworzenia pracy dyplomowej w  $\LaTeX$  dostarczając spójnych i konfigurowalnych rozwiązań dla różnych aspektów składu dokumentu.

## Rozdział 9

# Opracowanie szablonu pracy dyplomowej

Szablon jest stworzony pod wersje kompilatora Xe $\text{\LaTeX}$  . Pisząc pracę należy podjąć decyzje w jaki sposób będziemy pracować w sposób lokalny lub poprzez internetowy edytor Overleaf . Pisząc pracę lokalnie zapoznaj się z podrozdziałem ?? . Natomiast decydując się na Overleaf zapoznaj się z instrukcją poniżej. Używając Overleaf do pisania pracy należy rozwinąć "Menu" i zjechać do zakładki "Settings" i zmienić "Compiler" na Xe $\text{\LaTeX}$  ponieważ na starcie wybrany jest inny.



**Rysunek 1.** Ustawienie kompilatora na platformie Overleaf

Następnie należy wybrać z listy plików „Dyplom.tex” i ustawić typ swojej pracy w linii `\documentclass` zmieniając `[thesis=inz]` na jedną z dostępnych opcji `inz` (praca inżynierska w języku polskim), `mgr` (praca magisterska w języku polskim), `bsc` (praca inżynierska w języku angielskim) lub `msc` (praca magisterka w języku angielskim).

**Listing 4.** Ustawienia klasy dokumentu

```
8 \documentclass[thesis=inz]{TemplateCore/Dyplom}
9
10 thesis=[inz|mgr|bsc|msc]
11 * inz - praca inzynierska
12 * mgr - praca magisterska
13 * bsc - bachelor thesis
14 * msc - master thesis
```

---

Poniżej znajduje się kolejna ważna opcja ponieważ wpływa ona na rodzaj sortowania bibliografii w wierszu 19 zmieniając `sorting=nty` na dostępne opcje:

- `nty` (Nazwisko, Tytuł, Rok),
- `nyt` (Nazwisko, Rok, Tytuł),
- `none` (Bez sortowania).

**Listing 5.** Ustawienia sortowania bibliografii

```
19 \usepackage[sorting=nty, bibfile=Dyplom]{TemplateCore/Dyplom}
20
21 - nty (Nazwisko, Tytuł, Rok):
22 - nyt (Nazwisko, Rok, Tytuł):
23 - none (Bez sortowania):
24 - nty (Author, Title, Year):
25 - nyt (Author, Year, Title):
26 - none (No sorting):
```

---

Nazwa pliku przy `bibfile=` powinna być zgodna z nazwą pliku naszej bibliografii u autora jest to plik `Dyplom.bib`. Po wybraniu odpowiednich opcji należy uzupełnić pola do personalizacji strony tytułowej zaczynające się w wierszu 28 pliku `Dyplom.tex`.

**Listing 6.** Ustawienia dotyczące strony tytułowej

```
28 Konfiguracja - do personalizacji
29 Configuration - to be personalized
30
31 \kierunek{Informatyka}
32 \specjalnosc{Sieci}
33 \title{Szablom Pracy Dyplomowej}
34 \engtitle{Szablon Pracy Dyplomowej}
35 \album{99168}
36 \author{Wojciech Hyl}
37 \promotor{prof. dr hab. in\.z. Marcin Kowol}
38 \date{2023}
39 \longdate{2023-11-26}
```

---

---

W tym samym pliku po spersonalizowaniu należy napisać streszczenie w języku polskim jak i angielskim. Streszczenie powinno zawierać minimum 100 znaków do maksymalnie 4000 i należy je napisać w formie streszczenia a nie spisu treści W kodzie poniżej użyto pakiet `lipsum`, który generuje fikcyjny, standardowy tekst.

**Listing 7.** Miejsce pisania streszczenia pracy

```
47
48 Streszczenie pracy i abstract.
49 Summary of the Thesis and Abstract
50
51 \streszczeniepracy{
52
53
54     \lipsum[1-4]
55 } koniec streszczenia
56
57 \slowakluczowe{A, B, C}
58
59 \thesisabstract{
60
61
62     \lipsum[1-4]
63 } end of abstract
64
65 \thesiskeywords{X, Y, Z}
```

---

Główna struktura dokumentu została przedstawiona w listingu ??

**Listing 8.** Główna struktura dokumentu

```
1 % !TEX program = xelatex
2 % !TeX encoding = utf8
3 % !TeX spellcheck = pl-PL
4
5 % Wybierz rodzaj pracy dyplomowej
6 % Pick thesis type
7
8 \documentclass[thesis=inz]{TemplateCore/Dyplom}
9
10 % thesis=[inz|mgr|bsc|msc]
11 % * inz - praca inżynierska
12 % * mgr - praca magisterska
13 % * bsc - bachelor thesis
14 % * msc - master thesis
15
16 \usepackage[sorting=nty, bibfile=Dyplom]{TemplateCore/Dyplom}
```

---

```

17
18 %- nty (Nazwisko, Tytul, Rok):
19 %- nyt (Nazwisko, Rok, Tytul):
20 %- none (Bez sortowania):
21 % - nty (Author, Title, Year):
22 % - nyt (Author, Year, Title):
23 % - none (No sorting):
24
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 % Konfiguracja - do personalizacji
27 % Configuration - to be personalized
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 \kierunek{Informatyka}
30 %\specjalnosc{Sieci}
31 \title{Szablon \LaTeX\ do pracy dyplomowej i prezentacji}
32 }
33 \engtitle{Szablon \LaTeX\ do pracy dyplomowej i prezentacji}
34 }
35 \album{99168}
36 \author{Wojciech Hyl}
37 \promotor{prof. dr hab. in\.z. Marcin Kowol}
38 \date{2023}
39 \longdate{2023-11-26}
40
41
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
43 % Streszczenie pracy i abstract.
44 % Summary of the Thesis and Abstract
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 \streszczeniepracy{
47 \lipsum[1-4]
48 } % koniec streszczenia
49
50 \slovakluczowe{Sk Ćad tekstu \LaTeX, Praca Dyplomowa, Narz Żdzia i
   Żrodowiska  pracy}
51
52 \thesisabstract{
53 \lipsum[1-4]
54 } % end of abstract
55
56 \thesiskeywords{Text Composition \LaTeX, Dissertation, Tools and working
   environments.}

```



---

```

57 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59 % Tu zaczyna sie dokument
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61 \begin{document}
62   % Strony nagl wkowe
63   % Headers
64   \frontpages
65
66   % Wlasciwa tresc jest w pliku Chapters/main.tex
67   % Real contents is in Chapters/main.tex
68   \input{Chapters/main}
69
70   % Bibliografia - musi byc
71   % Bibliography - must exist
72   \bibliografia
73
74   % Strony koncowe - mozna zakomentowac, jesli zbedne
75   % Additional pages - comment out if not needed
76
77   \clearpage
78   % Spis rysunk w
79   \clearpage
80   \listoffigures
81   % Spis tabel
82   \clearpage
83   \listoftables
84
85   \clearpage
86   \lstlistoflistings
87
88
89 \end{document}
90 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

Po zakończeniu ustawiania i napisaniu streszczenia należy rozwinąć katalog Chapters i otworzyć plik `main.tex` gdzie można umieścić czy napisać poszczególne rozdziały pracy. Autor zaleca pisane rozdziałów w osobnych plikach ponieważ rozdziały napisane w jednym pliku będą bardzo duże i trudne do zarządzania. Chcąc dodać nowy rozdział należy utworzyć nowy plik w katalogu Chapters po czym w pliku `main.tex` i dodać go w odpowiednim miejscu. Trzeba pamiętać o tym że pliki umieszczone są w kolejności w jakiej będą prezentowane w pracy. W pliku `main.tex` żeby dodać nowy rozdział trzeba użyć poleceń `\chapters{Nazwa Rozdziału}` i `\input{ścieżka do pliku}`.

Listing 9. Tworzenie nowych rozdziałów

```

1
2 % Rozdziały zaczynają się od "chapter"
3 \chapter{Introduction}
4 % Praca podzielona na mniejsze pliki włączane za pomocą input
5 \input{Chapters/Introduction}

```

## 9.1 Rysunki

Rysunki  $\text{\LaTeX}$  umieszczamy w otoczeniu "figure" i za pomocą komendy `\includegraphics`. Otoczenie to pozwala na dodanie podpisu (`\caption`) i znacznika (`\label`) dzięki któremu potem można się do nich odnosić za pomocą komendy `\ref{sec:nazwa znacznika}`. Oto przykład wstawiania rysunku do dokumentu.

Listing 10. Kod wstawiający prosty rysunek

```

1 \begin{figure}[!hb]
2   \includegraphics{obraz.jpg}
3   \caption{Podpis do obrazu}
4   \label{fig:moja-etykieta}
5 \end{figure}

```

Rysunki można przenosić jeśli nie trafił tam gdzie powinien. Trzeba wtedy zmienić pozycję rysunku w przykładzie `[!hb]` na jedną z niżej wymienionych opcji:

- **h** - umieść rysunek tutaj (here), jeśli to możliwe,
- **t** - umieść rysunek na górze strony (top),
- **b** - umieść rysunek na dole strony (bottom).
- **p** - umieść rysunek na osobnej stronie (page).
- **!** - ignoruj większość zasad pozycjonowania i sugeruje, aby naruszyć pewne domyślne zasady.

Możesz również łączyć te opcje, na przykład `[ht]` oznacza "tutaj" lub na górze strony. Dodatkowe obcje których można użyć w otoczeniu `figure`

- **scale**  
Odpowiada za rozmiar, wprowadź `[scale=2.0]`. Ta liczba może przyjąć dowolną wartość; liczba między 0 a 1 zmniejszy grafikę, a liczba ujemna odwróci ją,
- **angle**  
Kąt jest podawany w stopniach i zgodnie z ruchem wskazówek zegara. Grafika jest obracana wokół swojego pochodzenia,
- **width**  
Umożliwia ustawienie szerokości obrazu,
- **height**  
Daje możliwość zmiany wysokości obrazu.



## Praca dyplomowa inżynierska

na kierunku Informatyka

Szablon  $\LaTeX$  do pracy dyplomowej  
i prezentacji

Wojciech Hyl  
numer albumu 99168

promotor  
prof. dr hab. inż. Marcin Kowol

Opole, 2023

**Rysunek 2.** Strona tytułowa pracy

Poniżej znajdziemy kod poprawnie wstawionego obrazu z przykładu ??

**Listing 11.** Kod poprawnie wstawionego obrazu

```

1 \begin{figure}[!h]
2   \includegraphics[width=0.5\linewidth]{IMAGE/Strona_tytulowa_pracy.
   png}
3   \caption{Strona tytułowa pracy}
4   \label{rys:Strona_tytulowa_pracy}
5 \end{figure}
```

## 9.2 Odwołania do literatury

Zarządzanie bibliografią w  $\LaTeX$  odbywa się automatycznie. Nie trzeba się przejmować dodawanie czy usuwaniem pozycji z bibliografii. Dzięki wcześniejszym ustawieniom bibliografia sortuje się według ustawionego schematu. W bibliografii pojawiają się tylko te przypisy, do których odwołania istnieją w treści. Odniesienia do literatury tworzy się za pomocą komendy `\cite{id}` w którym podaje się znacznik przypisany danej pozycji. Noty bibliograficzne powinny znajdować się w osobnym pliku

Dyplom.bib. Każda zmiana pliku bibliografii wymaga jego rekompilacji. W szablonie używany jest program biber który w Overleaf uruchamia się automatycznie przy zmianie bibliografii

Oto przykładowe odwołanie do literatury [8] \cite{fowler2009}.

Wszystkie odwołania robi się praktycznie tak samo. Chcąc odnieść się do kilku pozycji jednocześnie, trzeba ich znaczniki wpisać po przecinku [4, 8] \cite{kopka2003guide,fowler2009}.

Można też odnieść się do konkretnej strony [4, s. 38] \cite[s. 38]{kopka2003guide}.

Znak tyldy (~) przed wywołaniem cite „skleja” odnośnik z poprzedzającym je słowem za pomocą tak zwanej „twardej” spacji.

Ułatwieniem może być to, że serwisy takie jak Google Scholar, Mendeley czy Zotear pozwalają na pobranie cytowania w postaci BibTeX.

## 9.3 Tabele

Aby stworzyć prostą tabelę w  $\text{\LaTeX}$  możesz skorzystać z otoczenia table. Przykład tworzenia prostej tabeli w  $\text{\LaTeX}$

**Listing 12.** Kod prostej tabeli

```

1 \begin{table}[h]
2   \centering
3   \begin{tabular}{p{2.5cm}c|l}
4     Przykład & Przykład & Przykład\\ \hline
5     Przykład & Przykład & Przykład\\ \hline
6     Przykład & Przykład & Przykład \\ \hline
7     Przykład & Przykład & Przykład\\ \hline
8     Przykład & Przykład & Przykład\\ \hline
9     Przykład & Przykład & Przykład
10  \end{tabular}
11  \caption{\label{tab:Prosta tabela}Prosta tabela}
12 \end{table}
```

W przykładzie ?? został przedstawiony prosty kod tworzący table mieszczący się w otoczeniu table i tabular. Pozycje tabeli ustawiamy jak w przypadku rysunków ?? natomiast jeśli chodzi o otoczenie tabular przy którym specyfikacja kolumn określa liczbę i format kolumn tabeli. Liczbę kolumn i ułożenie w nich tekstu określa się jako argumenty polecenia tabular:

- l – dosunięcie do lewej,
- r – dosunięcie do prawej,
- c – centrowanie,
- p – styl akapitowy z szerokością określoną parametrem.

Wpisując zawartość każdej komórki tabeli, oddzielając je znakiem "&". Każdy wiersz kończ "\\". Komenda \hline służy do dodawania poziomych linii w tabelach. Jest to polecenie, które

umieszczasz wewnątrz środowiska tabeli. Kończąc tabele należy pamiętać aby nadać jej tytuł oraz etykietę. Tabela poniżej jest przykładem wygenerowanej tabeli z listingu ??.

**Tabela 1.** Tabela wygenerowana kodem z listingu 12

Przykład	Przykład	Przykład
Przykład	Przykład	Przykład
Przykład	Przykład	Przykład
Przykład	Przykład	Przykład
Przykład	Przykład	Przykład
Przykład	Przykład	Przykład

Tabele należą do najtrudniejszym elementem do wprowadzenia w  $\text{\LaTeX}$  ze względu na nagromadzenie znaczników. Dlatego warto rozważyć edytowanie tabel za pomocą dedykowanego edytora takiego jak TablesGenerator w których tworzymy tabele graficznie a edytor automatycznie generuje nam z tego kod  $\text{\LaTeX}$  który wystarczy wstawić w interesującym nas miejscu.

## 9.4 Wzory

Wzory matematyczne w  $\text{\LaTeX}$  zapisuje się za pomocą poleceń. Proste i popularne symbole są łatwe do zapamiętania. Natomiast, jeśli potrzebne są bardziej złożone wzory lub mniej powszechnie używane symbole, warto sięgnąć po dokumentację, dostępną na przykład na stronach takich jak Overleaf. Można skorzystać z edytorów równań dostępnych online, takich jak:

- CodeCogs
- Latex4technics

Ich zaletami są natychmiastowe generowane wzory. Wzory Matematyczne w  $\text{\LaTeX}$  piszemy w środowisku `equation` pisane wzory w tym środowisku są w tryb wyłączony (`display`) czyli występują w osobnej linii. Istnieje również możliwość pisanie wzorów matematycznych w Linii tekstu za pomocą znaków dolara. Poniżej znajduje się kilka przykładów zapisania wzorów. W pierwszym przykładzie użyto dodatkowej opcji otoczenia `equation` która wyłącza numerowanie poprzez dodanie do otoczenia `*`. Warto zapoznać się z dokumentacją dotyczącą wzorów na stronie internetowej [10]

$$r = \sqrt{r_x^2 + r_y^2}$$

$$\frac{dx}{dx} = 1 \tag{1}$$

$$\frac{d}{dx} \ln x = \frac{1}{x} \tag{2}$$

$$y = \varphi(y')x + \psi(y') \tag{3}$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} = \frac{1}{\det(X)} \begin{bmatrix} X_{22}Y_{11} - X_{12}Y_{21} & X_{22}Y_{12} - X_{12}Y_{22} \\ X_{11}Y_{21} - X_{21}Y_{11} & X_{11}Y_{22} - X_{21}Y_{12} \end{bmatrix} \quad (4)$$

## 9.5 Kody Źródłowe

Pakiet umożliwiający wstawianie kodów źródłowych z różnych języków programowania do dokumentu to `listing`. Więcej informacji można znaleźć na przykład w dokumentacji Overleaf. Poniżej znajdziesz krótką instrukcję używania pakietu `listing`:

Wstaw kod źródłowy do dokumentu, używając otoczenia `lstlisting`.

**Listing 13.** Przykładowy kod

```

1 \begin{lstlisting}[caption={Przykładowy kod}, label=lst:przyklad]
2 public class HelloWorld {
3     public static void main(String[] args) {
4         System.out.println("Hello, World!");
5     }
6 }
7 end{lstlisting}

```

**Listing 14.** Prosty program w języku python

```

1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 """Simple world of hello.
4 """
5
6 import sys
7
8 def main():
9     """The one and only function"""
10    fib = lambda n: reduce(lambda x, n: [x[1], x[0]+x[1]], range(n), [0,
11    1])[0]
12    try:
13        print(fib(int(sys.argv[1])))
14    except:
15        print("Hello World!")
16
17 if __name__ == "__main__":
18     main()

```

Kody źródłowe można też dołączać w zewnętrznych plikach i załączać automatycznie. Istnieje możliwość wskazania zakresu linii które pojawią się w dokumencie `listing` ?? ma automatycznie ustawiany podpis na podstawie nazwy pliku.

**Listing 15.** Kod pobrany z zewnętrznego pliku

```

33 for (i = 0; i < 120; i++) {
34     printf("*");

```

```

35 };
36
37 printf("Autor Programu: Wojciech Hyl\n");
38
39 for (i = 0; i < 120; i++) {
40     printf("*");
41     srand((unsigned int)time(NULL));
42 };
43 do {
44     n++;
45     c = 0;
46     for (i = 0; i < 120; i++) {
47         printf("*");
48     };
49
50     printf("Zadaj przedzia ̇ losowanych liczb\n");

```

## 9.6 Grafy, schematy, rysunki techniczne

Pakiet tikz w  $\text{\LaTeX}$  umożliwia rysowanie grafiki wektorowej i tworzenie różnych rodzajów ilustracji, w tym diagramów, grafów, schematów, czy rysunków technicznych. W tworzeniu schematów może pomóc internetowy edytor Tikzmaker który generuje kod na podstawie grafiki.

Poniżej znajdziesz podstawową instrukcję dotyczącą korzystania z otoczenia tikzpicture:

Umieść otoczenie tikzpicture wewnątrz środowiska figure, jeśli chcesz mieć możliwość dodania podpisu i etykiety. Poniżej w listingu 17 został przedstawiony przykład rysujący funkcję kwadratową z rysunku ??.

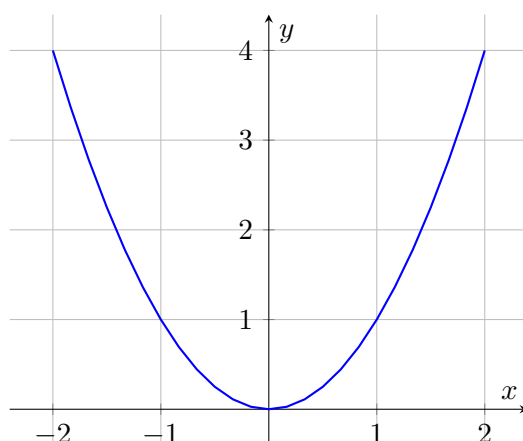
**Listing 16.** Kod rysujący funkcję kwadratową

```

1 \begin{figure}
2   \begin{tikzpicture}
3     \begin{axis}[
4       xlabel=$x$,
5       ylabel=$y$,
6       grid=both,
7       axis lines=middle,
8       enlargelimits,
9     ]
10    \addplot[domain=-2:2, color=blue, thick]{x^2};
11  \end{axis}
12 \end{tikzpicture}
13 \caption{Funkcja kwadratowa}
14 \label{fig:Funkcja kwadratowa}
15 \end{figure}

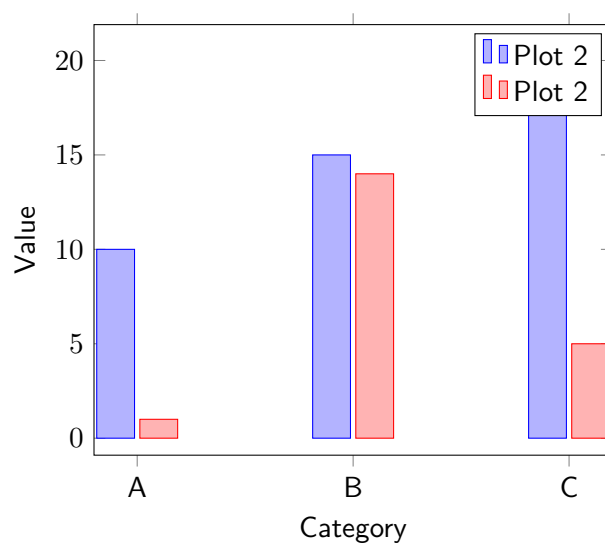
```





**Rysunek 3.** Funkcja kwadratowa z listingu 16

Szablon umożliwia tworzenie wykresów poprzez wczytywanie danych z plików. Dane do wykresów powinny być umieszczone w folderze `charts`. Wykres ?? został stworzony poprzez wczytanie danych z pliku za pomocą kodu podanego w listingu ??



**Rysunek 4.** Przykład wykresu wczytywanego z pliku

Listing 17. Wykrez wczytany z pliku

```

1 \begin{figure}[!hb]
2   \centering
3   \begin{tikzpicture}
4     % First subplot
5     \begin{axis}[
6       ybar,
7       xlabel={Category},
8       ylabel={Value},
9       bar width=0.5cm,
10      symbolic x coords={A, B, C},
11      xtick=data,
12    ]
13      \addplot table [x index=0, y index=1] {charts/bar_data1.txt};
14      \addlegendentry{Plot 2}
15      \addplot table [x index=0, y index=1] {charts/bar_data3.txt};
16      \addlegendentry{Plot 2}
17    \end{tikzpicture}
18    \caption{Multiplot Example}
19    \label{fig:multiplot}
20  \end{figure}
21

```

Dokumentacja TikZ jest obszerna i zawiera wiele przykładów. Warto ją przeczytać, aby poznać pełne możliwości pakietu. TikZ oferuje wiele zaawansowanych funkcji, a dokładne dostosowania można znaleźć w jego dokumentacji. [tikz.dev](#)

Pakiet circuitikz umożliwia rysowanie obwodów elektrycznych. Oto podstawowe instrukcje dotyczące korzystania z circuitikz: Rozpoczęcie Środowiska circuitikz: Umieść otoczenie circuitikz wewnątrz otoczenia figure, jeśli chcesz mieć możliwość dodania podpisu i etykiety. Poniżej w lisnigu ?? został przedstawiony kod rysujący schemat przedstawiony w schemacie ??

Listing 18. Schemat użycia circuitikz

```

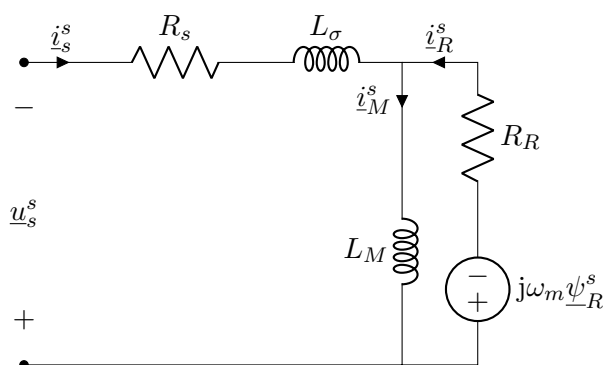
1 \begin{figure}[H]
2   \centering
3   \begin{circuitikz}[american voltages]
4     \draw
5       (0,0) to [short, *-] (6,0)
6       to [V, l_=${\mathrm{j}}{\omega}_m \underline{\psi}^s_R$] (6,2)
7       to [R, l_=${R}_R$] (6,4)
8       to [short, i_=${\underline{i}}^s_R$] (5,4)
9       (0,0) to [open, v^>=${\underline{u}}^s_s$] (0,4)
10      to [short, *- ,i=${\underline{i}}^s_s$] (1,4)
11      to [R, l=${R}_s$] (3,4)
12      to [L, l=${L}_{\sigma}$] (5,4)

```

```

13   to [short, i_=$\underline{i}^s_M$] (5,3)
14   to [L, l_=$L_M$] (5,0);
15   \end{circuitikz}
16   \caption{Schemat}
17   \label{fig:funkcja}
18   \end{figure}

```



Rysunek 5. Schemat

Dokumentacja: Pełna dokumentacja circuitikz zawiera obszerny spis elementów i dostępnych opcji. Można ją znaleźć na [Circuitikz Dokumentacja](#)

## 9.7 Dodatki

### 9.7.1 Podstawowe style tekstu

- **tekst wytłuszczony** `\textbf{tekst wytłuszczony}`,
- *tekst pochylony* `\textit{tekst pochylony}`,
- tekst maszynowy `\texttt{tekst maszynowy}`,
- tekst podkreślony `\underline{tekst podkreślony}`,
- W pracy zostało stworzone makro które zmienia kolor tekstu maszynowego na szary `\maszynowy{tekst maszynowy}`.

### 9.7.2 Wiszące znaki

W polskiej tradycji literackiej pojedyncze znaki, takie jak a, i, o, u, w, z, nie powinny występować na końcu linii. Aby uniknąć tego problemu i połączyć tę literę z następującym słowem, używa się znaku tyldy (`~`).

Na koniec pisania pracy najlepiej zamienić wszystkie ciągi znaków przy pomocy „znajdź i zamień” z `Space` + `znak` + `Space` na `Space` + `znak` + `~`.

### 9.7.3 Wyliczenia

Tworzenie wyliczeń w  $\text{\LaTeX}$  odbywa się za pomocą otoczenia `itemize` w którym każdy kolejny punkt dodajemy za pomocą komendy `\item`. Poniżej znajduje się przykład użycia otoczenia `itemize`

- standardową "kropkę",
- myślnik.

Podczas używania warto poznać zasady interpunkcji w wyliczeniach z którymi można zapoznać się w artykule [7] lub w blogu [11].

### 9.7.4 Akronimy i symbole

Szablon używa pakietu `glossaries` do zarządzania akronimami i symbolami. Listę tych elementów należy przygotować w pliku `glossary.tex`, zgodnie z pokazanym szablonem `\newacronym{PO}{{PO}}{Politechnika Opolska}`.

### 9.7.5 Ozdobniki graficzne w opisie oprogramowania

Szablon wczytuje pakiet pozwalający na wyróżnienie w tekście informacji o skrótach klawiszowych, poruszaniu się po menu programu i ścieżki plików. Aby użyć tego pakietu należy zastosować komendę `\keys{Nazwa klawisza}`.

W tekście można wyróżnić skróty klawiszowe, takie jak na przykład:

- `Alt` + `F4`, `Ctrl` + `Alt` + `Del.`,
- `A` + `a` + `B` + `b` + `C` + `c` + `1` + `2` + `3` + `PgUp`,
- `Space`,
- `←` `Del.` `Del.`,
- `↵` `Enter`,
- `↑` `↓`.

### 9.7.6 Kompilator lokalny

Problemem darmowej wersji Overleaf jest to, że zapewnia krótki czas kompilowania nie dłuższy niż 1 minute. W przypadku gdy praca jest skomplikowana czas ten może zostać przekroczony i nie wygeneruje nam pliku wynikowego. Rozwiązaniem tego problemu jest zainstalowanie na własnym komputerze kompilatora lokalnego. Aby to poprawnie zrobić najpierw należy zainstalować Ghostscript. Ghostscript to interpreter języka PostScript i Portable Document Format (PDF), służący do przetwarzania i wyświetlania plików w tych formatach. Po udanym zainstalowaniu Ghostscript należy zainstalować MiKTeX który jest dystrybucją oprogramowania do składu tekstu opartą na  $\text{\TeX}$  /  $\text{\LaTeX}$ . Ostatnim wymaganym programem jest edytor tekstu który możemy wybrać według własnych preferencji. Najczęściej używane edytory tekstu to:

- TeXworks,
- Texstudio,

- WinEdt,
- Visual Studio Code do którego należy dodać rozszerzenie `LaTeX Workshop`

Dzięki dodaniu na początku dokumentu tzw. komentarzy magicznych (ang. magic comments) lub dyrektywy `TeX` (ang. `TeX directives`) edytory tekstu automatycznie pobierają informacje jakiego kompilatora powinny użyć, w jakim kodowaniu jest dokument i włączają sprawdzanie pisowni w języku polskim.

**Listing 19.** Dyrektywy `TeX`

```
1 % !TeX program = xelatex
2 % !TeX encoding = utf8
3 % !TeX spellcheck = pl-PL
```

---

### 9.7.7 Oświadczenia i wnioski

Wyborze plików po lewej stronie również są wrzucone oświadczenia i wnioski które mogą się przydać podczas składania pracy.

## 9.8 Prezentacja

Podczas tworzenia szablonu do pracy dyplomowej został stworzony szablon do prezentacji dyplomowej. Wygląd prezentacji został oparty na podstawie zatwierdzonego szablonu przygotowanego w Microsoft PowerPoint. Do tworzenia prezentacji w `LaTeX` istnieje osobna klasa dokumentu `beamer`. Używając w ten sposób klasy tworzy nam się prezentacja w rozdzielczości 4:3 dzięki argumentowi `[aspectratio=169]` jesteśmy w stanie zmienić rozdzielczość naszej prezentacji na 19:4. W stworzonej przez autora prezentacji istnieje kustomizowany slajd tytułowy w którym automatycznie zostają umieszczone tytuł, podtytuł, autor, oraz instytucja podane w odpowiednich miejscach.

**Listing 20.** Personalizacja prezentacji

```
1 \title{Szablon prezentacji \LaTeX}
2 \subtitle{Podtytuł}
3 \author[Autor]{Autor}
4 \institute{Instytucja}
```

---

Nowe slajdy w prezentacji tworzy się za pomocą otoczenia `frame` w lizingu ?? przykład tworzenia nowego slajdu

**Listing 21.** Tworzenie pustego slajdu w prezentacji

```
1 \begin{frame}
2   \frametitle{Tytuł slajdu}
3   Treść slajdu
4 \end{frame}
```

---

W miejscu `frametitle` należy wpisać tytuł slajdu. W następnym wierszu umieszczamy treść naszego slajdu i całość zamykamy otoczeniem `frame`. Prezentacja posiada pakiety takie same jak szablon do pracy dyplomowej dzięki czemu tworzenie nowych slajdów i dodawanie do nich takich elementów jak wzory, tabele czy rysunki odbywa się w ten sam sposób jak przedstawiony w podrozdziałach ??, ??, ???. Jednak należy pamiętać aby dodać listing do prezentacji trzeba użyć dodatkowego argumentu `[fragile]` przy tworzeniu nowego slajdu.

**Listing 22.** Slajd z kodem źródłowym

```
1 \begin{frame}[fragile]
2 \frametitle{Prezentacja w \LaTeX\ }
3 Fragment kodu
```

---

W prezentacjach  $\text{\LaTeX}$  jest możliwość automatycznego podziału długich zawartości slajdów na kilka stron za pomocą opcji `allowframebreaks`. Jest szczególnie przydatna, gdy zawartość jednego slajdu nie mieści się na ekranie i potrzebny jest automatyczny podział na kolejne strony. Ta opcja pomaga w utrzymaniu czytelności i przejrzystości prezentacji.



## Szablon prezentacji $\text{\LaTeX}$

Podtytuł

Autor  
Instytucja

**Rysunek 6.** Strona tytułowa prezentacji

## Rozdział 10

# Podsumowanie

W celach realizacji szablonu do pracy dyplomowej i prezentacji podjęto następujące kroki :

- przeprowadzona analizę wymagań dotyczących pracy dyplomowej uwzględniając w niej wytyczne uczelni. Zwarto wszystkie elementy które powinny znajdować się w szablonie.
- dostosowanie szablonu do oficjalnych wymagań co do formatowania, struktury i prezentacji pracy dyplomowej.
- wybrano odpowiednią klasę dokumentu  $\text{\LaTeX}$  tak aby najlepiej dopasowała się do wymagań pracy dyplomowej pod względem formatowania i struktury dokumentu.
- sporządzono strukturę dokumentu uwzględniając kolejność rozdziałów, miejsce spisu treści, bibliografii i innych elementów pracy .
- dostosowanie style tekstu, ustalając fonty, marginesy, interlinię, oraz inne parametry. formatowania tak aby były zgodne z wymaganiami uczelni a zarazem tworzyły spójny wygląd
- uruchomienia wsparcia dla elementów graficznych takich jak rysunki, tabele czy schematy Zapewniono aby elementy były zgodne z stylem pracy.
- zaimplementowano nagłówki i stopki żeby zawierały niezbędne informacje takie jak nazwa i numer rozdziału, numeracja stron, itp.
- implementacja przypisów umożliwiającą dodawanie przypisów Górnych i dolnych
- przeprowadzono test szablonu żeby sprawdzić jego poprawność formatowania i zgodność z wytycznymi uczelni.
- przygotowano przewodnik dla użytkownika szablonu zawierający takie informacje jak dotyczące obsługi, dostosowań, itp.

Podjęcie tych kroków pozwoliło na stworzenie profesjonalnie wyglądającego szablonu spełniającego wymagania uczelni oraz polskiej tradycji piśmienniczej. Ułatwiło również Studentom proces tworzenia profesjonalnych dokumentów dyplomowych.

# Bibliografia

- [1] **Johanna Cleary**, „Creating Bibliography with Latex,” *Journal of Radio*, t. 79, 2 2004.
- [2] **A. Diller i J. Jełowicki**, *LaTeX wiersz po wierszu: zasady i techniki przetwarzania dokumentów*. Wydawnictwo Helion, 2001, ISBN: 9788371973413.
- [3] **Donald Knuth**, „The TEXbook Addison–Wesley,” *Reading, MA*, 1996.
- [4] **Helmut Kopka i Patrick W Daly**, *Guide to LATEX*. Pearson Education, 2003.
- [5] **Jerzy Kucharczyk**, *Wprowadzenie do systemu komputerowego składu tekstów drukarskich LATEX*. Wydawn. Uniwersytetu Wrocławskiego, 1994.
- [6] **Leslie Lamport**, „LATEX–A Document Preparation System Addison–Wesley,” *Reading, MA*, 1985.
- [7] **Łukasz Mackiewicz**. „Interpunkcja w wyliczeniach: przecinki, średniki, a może nic?” (2024), adr.: <https://www.ekorekta24.pl/interpunkcja-w-wyliczeniach-przecinki-sredniki-a-moze-nic/>.
- [8] **Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle i Chris Rowley**, *The LATEX companion*. Addison-Wesley Professional, 2004, ISBN: 978-0750685672.
- [9] **Tobias Oetiker** i in., *Nie za krótkie wprowadzenie do systemu LATEX2 $\epsilon$* , 2022.
- [10] **Overleaf**, *Overleaf - Online LaTeX Editor*, 2024. adr.: <https://www.overleaf.com/>.
- [11] **Justyna Szymczyk-Mielniczyn**. „Wyliczenia w tekstach.” Dostęp z dnia 26 stycznia 2024 roku. (), adr.: <https://contentwriter.pl/wyliczenia-w-tekstach/>.
- [12] **Claudio Vellage**. „Bibliography in LaTeX with Bibtex/Biblatex.” (2017), adr.: <https://latex-tutorial.com/tutorials/bibtex/>.



## **Spis rysunków**

## **Spis tabel**

# Listings

1	Przykład prostego dokumentu . . . . .	9
2	Przykłady dodawania do bibliografii . . . . .	16
3	Dostosowywanie nazwa do języka polskiego . . . . .	18
4	Ustawienia klasy dokumentu . . . . .	21
5	Ustawienia sortowania bibliografii . . . . .	22
6	Ustawienia dotyczące strony tytułowej . . . . .	22
7	Miejsce pisanie streszczenia pracy . . . . .	23
8	Główna struktura dokumentu . . . . .	23
9	Tworzenie nowych rozdziałów . . . . .	26
10	Kod wstawiający prosty rysunek . . . . .	26
11	Kod poprawnie wstawionego obrazu . . . . .	27
12	Kod prostej tabeli . . . . .	28
13	Przykładowy kod . . . . .	31
14	Prosty program w języku python . . . . .	31
15	Kod pobrany z zewnętrznego pliku . . . . .	31
16	Kod rysujący funkcje kwadratową . . . . .	32
17	Wykres wczytany z pliku . . . . .	34
18	Schemat użycia circuitikz . . . . .	34
19	Dyrektywy T <sub>E</sub> X . . . . .	37
20	Prsonalizacja prezentacji . . . . .	37
21	Tworzenie pustego slajdu w prezentacji . . . . .	37
22	Slajd z kodem źródłowym . . . . .	38