Implementasi Pemrograman Berorientasi Objek Pada Aplikasi Persuratan Sederhana Menggunakan Bahasa Pemrograman Java

1st Sudirman Sudirman
Teknologi Informasi
Universitas Bosowa
Makassar, Indonesia
sudirman.dymand@universitasbosowa.ac.id

2nd Uswatun Hasanah
Teknologi Informasi
Universitas Bosowa
Makassar, Indonesia
uthasanah24@gmail.com

3rd Andika Putra Ramadhani
Teknologi Informasi
Universitas Bosowa
Makassar, Indonesia
andikaputraramadhani232@gmail.com

***Abstract*— Perubahan sistem kerja pada era digital dan menjadikan semua pekerjaan harus berganti mengikuti perubahan teknologi yang ada juga mengubah pola atau aturan kerja pekerjaan yang sebelumnya dilakukan tanpa memanfaatkan teknologi kini haruslah disesuaikan dengan teknologi. Termasuk pada sistem yang bekerja dalam memberikan banyak informasi atau komunikasi dalam segala arah dan bidang. Dalam penerapan sistem informasi di bidang persuratan, aksesibilitas paket surat dapat dilakukan secara transparan. Disebut transparan dikarenakan pada aplikasi pengaksesan dapat dilihat oleh pimpinan bukan hanya oleh staf atau administrasi. Sistem informasi yang bekerja pada masalah persuratan menjadikan perubahan dari proses surat menyurat manual atau konvensional menjadi modern dan lebih canggih.**

Kata Kunci : Persuratan, Sistem Informasi Persuratan, Pemrograman Berorientasi Objek.

I. INTRODUCTION

Perkembangan teknologi Informasi terus mengalami kemajuan di segala aspek kehidupan, begitu banyak instansi baik pemerintah maupun swasta mulai menerapkan teknologi informasi untuk mengoptimalisasi segala proses, termasuk dalam sistem birokrasi yang bersifat administratif seperti surat menyurat.

Analisa dan disain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artefak dari proses analisis dan disain berorientasi objek adalah Unified Modelling Language (UML).

Sejauh ini, penelitian pengembangan suatu perangkat lunak dengan pendekatan paradigma berorientasi objek yang menggunakan UML sebagai tool untuk analisa dan disain telah banyak dilakukan, diantaranya analisa dan disain untuk memodelkan basis data (Sparks, 2001), memodelkan proses bisnis (castela, dkk., 2000) dan untuk merepresentasikan skema XML (Bernauer, dkk., 2004).

IBM telah mengembangkan sistem automatisasi dilingkungan kerja perkantoran berbasiskan teknologi informasi (Gardner, 2003). Sistem ini dikenal sebagai OFS (Office System), merupakan aplikasi dasar V1370 yang mampu menangani dan melayani seluruh sistem persuratan internal perkantoran. Aspek-aspek yang mampu diakomodasikan oleh sistem ini meliputi pembuatan konsep surat, pengarsipan surat, penyimpanan surat dan mendistribusikan surat ke tujuan surat.

II. PROBLEM STATEMENT

Berdasarkan Permasalahan tersebut dalam jurnal ini akan di bangun sebuah aplikasi untuk membantu merancang aplikasi surat masuk dan surat keluar, aplikasi tersebut dapat membantu melaukan pencarian arsip persuratan dengan lebih pratis dan cepat.

2.1. Literature Riview

Pada penelitian [1] dari Arfhan Prasetyo Judul" Rancangan Pembelajaran Pemrograman Java Dengan Pendekatan Object-First Untuk Mempermudah Memahami Konsep Object Oriented Programing (Oop)". Menggunakan metode analisis T-Test menguji kesamaan rata-rata dari 2 populasi yang bersifat independen (populasi yang satu tidak dipengaruhi atau tidak berhubungan dengan populasi yang lain) dengan melakukan pengumpulan data dengan melakukan penyebaran kuisioner sebelum dan sesudah metode pembelajaran dengan pendekatan object-first diterapkan.

Pada penelitian [2] dari Hermansyah, Een Juhriah, Maria Adelina Saragih Judul "Perancangan Aplikasi Surat Masuk dan Surat Keluar Bebasis java di PT Afconsult Energy Indonesia". Menggunakan metode grounded (grounded research) yaitu suatu metode penelitian berdasarkan pada fakta dan menggunakan analisis perbandingan dengan tujuan mengadakan generalisasi empiris, menetapkan konsep, membuktikan teori, mengembangkan teori, pengumpulan dan analisis data dalam waktu yang bersamaan. Setelah mengumpulkan data, peneliti melanjutkan proses penelitian sesuai dengan langkah-langkah pokok yang digunakan pada metode ini. Metode pengumpulan data yang dilakukan oleh peneliti untuk mendapatkan data-data serta informasi.

Pada Penelitian [3] dari Gusti Purnama Sari, Jefri Marzal, dan Mauladi Judul "Rancang Bangun Sistem Informasi Persuratan dan Disposisi Elektronik Universitas Jambi". Menggunakan metode Rapid Application Development (RAD). RAD merupakan gabungan dari bermacam-macam teknik terstruktur dengan teknik prototyping dan teknikpengembangan joint application untuk mempercepat pengembangan system.

Pada penelitian [4] dari Afftario Fauqa Wicaksana, Augie David Manuputty Judul "Perancangan Sistem Informasi Persuratan Berbasis Desktop Di Bagian Sekretariat Dinas Pekerjaan Umum Kota Semarang". Menggunakan metode penyusunan yang sistematis untuk mempermudah langkah-langkah yang akan diambil Kemudian dilakukan

pengujian sistem yaitu dengan melakukan pengujian sistem yang telah dibuat ke dalam perangkat yang berbasis desktop bahwa sudah siap untuk dioperasikan.

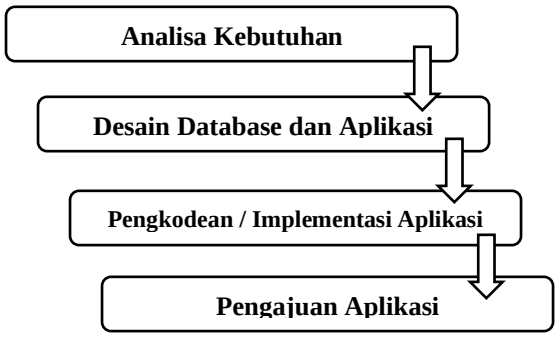
Pada penelitian [5] dari Gusti Agung Ayu Putri Judul "Rancang Bangun Sistem Informasi Persuratan Dan Kearsipan Universitas Udayana Menggunakan Paradigma Pemrograman Berorientasi Objek". Menggunakan paradigma pemrograman berorientasi objek dengan tahapan-tahapan pengembangan hingga Tahapan akhir dari penelitian ini adalah tahapan pengujian, yaitu menguji prototype yang telah dibuat menggunakan metode black box, yaitu pengujian dilakukan dengan memasukkan suatu input dan memeriksa apakah output yang dihasilkan sesuai dengan apa yang direncanakan.

III. METHODOLOGY

3.1. Metode Perancangan Sistem

Pengembangan sistem metodologi yang digunakan adalah *System Development Life Cycle* (SDLC) yang sering disebut dengan model pengembangan air terjun (waterfall). Metode waterfall merupakan model pengembangan sistem informasi yang sistematik dan sekuensial (Sasmito, 2017).

Metode penelitian yang digunakan untuk pengembangan perangkat lunaknya yaitu menggunakan model air terjun (waterfall). Model ini merupakan pendekatan perangkat lunak secara terurut yang dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (Rosa dan Shalahudin, 2013).



Gambar 1. Kerangka Penelitian

Analisis Kebutuhan yakni untuk menganalisis kebutuhan apa saja yang nantinya akan dibutuhkan pada saat pembuatan program, misalnya *hardware* dan *software* yang akan digunakan. Pada kasus ini menggunakan bahasa pemrograman java dengan Netbeans 8.2. Untuk desain *database* dan aplikasi, yakni memetakan apa saja kolom dan atribut yang digunakan dalam aplikasi serta menganalisis class dan method dalam aplikasinya.

Tahap pengkodean atau implementasi aplikasi menggunakan penerapan dari OOP atau *Objected Oriented Programming*. Tahap terakhir adalah tahap pengujian aplikasi, dimana aplikasi tersebut digunakan dan dicek apakah bisa berjalan atau tidak, ada error atau tidak. Sehingga keluaran dari program tersebut dapat bermanfaat. Model pengujian pada penelitian ini yaitu menggunakan angket dalam skala likert yang dilakukan pada 8 orang mahasiswa. Skor yang digunakan dalam rentang 0 sampai 5 yang ditunjukkan pada Tabel 2, serta aspek penilaian yang dievaluasi ditunjukkan pada Tabel 3.

Tabel 2. Skala Likert

Kriteria	Skor
Sangat Baik	5
Baik	4
Cukup	3
Kurang	2
Sangat Kurang	1

Tabel 3. Aspek Penilaian

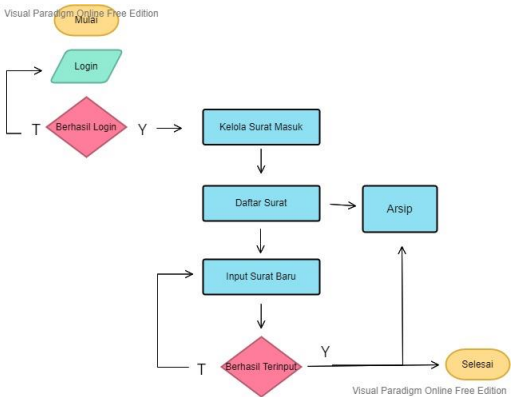
No	Aspek Penilaian
1	Operasional Sistem Aplikasi
2	*User Interface* Aplikasi

Tabel 4. Kriteria Interpretasi Skor Berdasarkan Interval Skala Likert

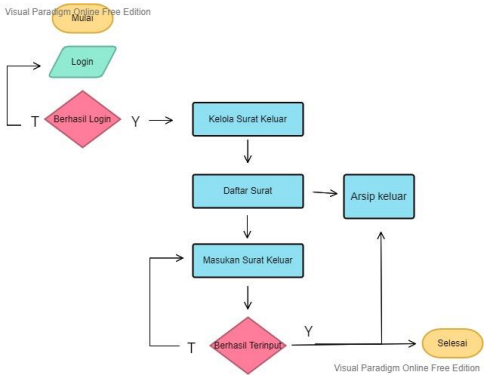
Kriteria	Keterangan
Angka 0% – 19,99%	Tidak setuju / buruk tidak
Angka 20% – 39,99%	Setuju / Kurang Baik
Angka 40% – 59,99%	Cukup/Netral
Angka 60% – 79,99%	Setuju / baik
Angka 80% – 100%	Setuju / sangat baik

3.2. Flowchart

Dalam pengembangan aplikasi ini peneliti membuat alur berupa Flowchart sebagai dasar pengembangan system ini. Dan dalam Flowchart ini terdapat 2 macam yaitu Flowchart untuk alur surat masuk dan keluar. Seperti dibawah ini:



Gambar 1 Flowchart Surat Masuk



Gambar 2 Flowchart Surat Keluar

IV. RESULT

4.1 Database

Pada bagian ini akan ditampilkan struktur database yang digunakan dalam pengembangan aplikasi ini.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	no_surat	int(11)			Tidak	*Tidak ada*		AUTO_INCREMENT
2	tanggal_terima	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
3	terima_dari	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
4	isi_prihal	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
5	lampiran	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
6	keterangan	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		

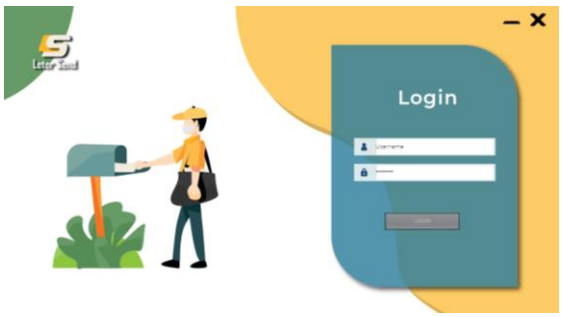
Gambar 3 Struktur Database Surat Masuk

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	no_surat	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
2	tanggal	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
3	isi_prihal	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
4	tujuan	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		
5	lampiran	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*		

Gambar 4 Struktur Database Surat Keluar

4.2. Jalannnya Program dan Penerapan GUI

Pada tahapan pertama pengguna akan dihadapkan pada menu utama, pada menu utama tersebut terdapat menu login untuk dapat masuk kedalam aplikasi persuratan.

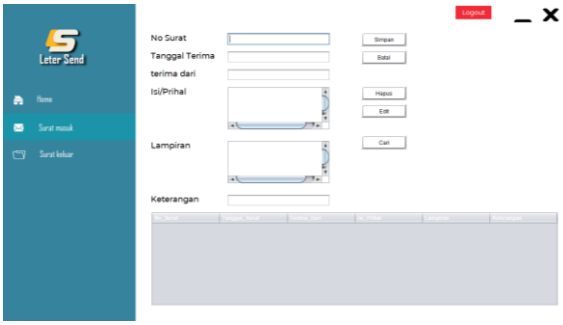


Gambar 5. Halaman Login

Lalu selanjutnya, dari tampilan login mengarah pada tampilan beranda setelah memasukkan username untuk login. Sehingga muncul tampilan seperti berikut.

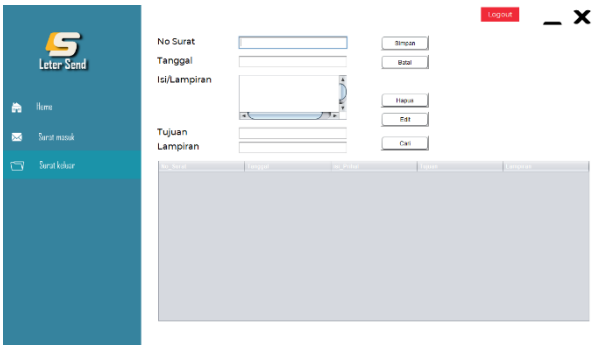
Gambar 6. Halaman Home

Pada tampilan ini admin dapat mengimput data surat masuk. Juga dapat mencari surat masuk yag terdapat dalam databasenya.



Gambar 7. Halaman Surat Masuk

Sebaliknya dari tampilan surat masuk, pada tampilan ini admin dapat mengimput data surat keluar, mengirim surat yang terdapat dalam databasenya.



Gambar 8. Halaman Surat Keluar

4.3. Penerapan OOP

- Penerapan Tipe Data

Ada beberapa tipe data yang digunakan pada program ini, contohnya adalah String yang digunakan untuk memasukkan karakter dan Integer. Tipe data yang digunakan didalam program ini ditunjukkan pada source code :

```
int baris = jTable1.rowAtPoint(evt.getPoint());
String no_surat =jTable1.getValueAt(baris, 1).toString();
no1.setText(no_surat);
String tanggal = jTable1.getValueAt(baris,2).toString();
tgl1.setText(tanggal);

String isi = jTable1.getValueAt(baris, 3).toString();
isi1.setText(isi);
String tujuan=jTable1.getValueAt(baris, 4).toString();
tujuan1.setText(tujuan);
String lampiran = jTable1.getValueAt(baris, 5).toString();
lampir1.setText(lampiran);
```

Gambar 10. Penerapan Tipe Data

- Penerapan Kelas dan Objek

OOP tidak bisa terlepas dari kelas dan objek yang digunakan. Tentunya hal ini sebagai langkah untuk membuat suatu sistem berbasis aplikasi yang bisa terintegrasi dan terhubung antara fitur satu dan fitur lainnya. Berikut adalah penerapan Kelas terdapat pada *source code:*

```
public class koneksi {

public class Surat_KM {
```

Gambar 11. Penerapan Kelas

```
login lg = new login();
```

Gambar 12. Penerapan Objek

- Penerapan Enkapsulasi

Salah satu penerapan enkapsukasi pada program ini digunakan untuk memberikan hak akses pada beberapa tombol, sehingga tidak terdapat 2 fungsi dalam satu tombol. Hal ini sangat dibutuhkan untuk membangun sebuah aplikasi yang memilki banyak fitur di setiap halamannya, seperti aplikasi ini. Penerapan enkapsulasi terdapat pada *source code:*

```
// Variables declaration - do not modify
private javax.swing.JLabel close;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JPanel jPanel7;
private javax.swing.JLabel minimize;
// End of variables declaration
```

Gambar 13. Penerapan Enkapsulasi

- Penerapan *Inheritance*

Class yang mengandung member yang sama dari beberapa *class* lain dinamakan *superclass* atau *parent class. Class* yang mewarisi dinamakan *subclass* atau *child class. Inheritance* menghasilkan *class hierarchy* yang memudahkan untuk menghubungkan halaman tampilan GUI dengan kelas-kelas fitur yang berisi metode-metode setiap tampilan pada aplikasi tersebut terhubung. Pada program ini penerapan *Inheritance* terdapat pada *source code:*

```
public class Home extends javax.swing.JFrame {

public class login extends javax.swing.JFrame {

 public class surat_keluar extends javax.swing.JFrame {

public class surat_masuk extends javax.swing.JFrame {
```

Gambar 14. Penerapan *Inheritance* pada Aplikasi

- Penerapan *Polymorphism*

Polymorphism artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai polymorphism, bila objek-objek itu mempunyai antar muka yang identik namun mempunyai perilaku yang berbeda. Hal ini sangat berguna untuk membuat suatu aplikasi lebih efisien, karena tidak perlu membuat objek baru lebih banyak untuk menyesuaikannya dengan sebuah *method*. Pada program ini penerapan *Polymorphism* terdapat pada *source code:*

```
public void mouseClicked(java.awt.event.MouseEvent evt) {
    jPanel6MouseClicked(evt);
}
public void mousePressed(java.awt.event.MouseEvent evt) {
    jPanel6MousePressed(evt);
}
```

Gambar 15. Penerapan *Polymorphism* pada Aplikasi

- Penerapan *Interface Class*

Interface pemrogram, untuk berbagi konstanta atau menentukan bentuk metode yang dapat digunakan oleh sejumlah kelas. *Interface* diimplementasikan ke dalam suatu *class* dengan menggunakan kata kunci *implements.* Sebuah kelas dapat mengimplementasikan lebih dari satu *interface*. *Interface* berfungsi untuk mendeklarasikan terlebih dahulu method-method yang ingin dibuat. Jika ingin memakai method tersebut di kelas lain, hanya perlu melakukan pemanggilan dan *implements.* Pada program ini penerapan *Interface* terdapat pada *source code:*

```
public void run() {
    new login().setVisible(true);
}
```

Gambar 16. Penerapan Interface Class

- Penerapan *Method Mutator*

Method Mutator adalah *Method* yang digunakan untuk akses objek lain agar bisa mengubah data. *Method* ini biasanya sering digunakan ketika membuat sebuah program pendataan, karena banyak inputan yang sama dan sedikit yang berbeda. Seperti pada aplikasi ini yang menginputkan nomor polisi dan jenis kendaraan. Pada umumnya penulisan *method* ini biasanya diawali dengan penulisan set.

```
no.setText("");
tgl.setText("");
terima.setText("");
isi.setText("");
lampir.setText("");
ket.setText("");
```

Gambar 17. Penerapan *Method Mutator*

- Penerapan *Exception Handling*

Exception Handling adalah *event* yang terjadi ketika program menemui kesalahan pada saat instruksi program dijalankan. Pada program ini mendeteksi apabila terjadi kesalahan ketika menginputkan dan menyimpan data pada *database*.

```
public void Connection(){
    try {
        con=DriverManager.getConnection("jdbc:mysql://localhost/surat", "root", "123456789");
        stm = con.createStatement();

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "koneksi gagal "+e.getMessage());
    }
}
```

Gambar 18. Penerapan *Exception Handling*

4.4. Hasil Pengujian Aplikasi

Pada penelitian ini, dilakukan uji coba pengguna kepada 8 responden. Data hasil uji coba diperoleh dengan cara menyebarkan angket pertanyaan beserta skala penilaiannya. Hasil uji coba Pengguna disajikan pada Tabel 4 dan Tabel 5.

Table 4. Hasil Uji Coba Pengguna

Indicator Penilaian	Hasil responden				
	1	2	3	4	5
Tampilan Awal	0	0	0	0	8
Fitur Surat	0	0	0	2	6
Cara Kerja (Operasional) Aplikasi	0	0	0	2	6
Tampilan	0	0	0	0	8
Keseluruhan / *Interface* Aplikasi	0	0	0	1	7

Table 5. Presentase Uji Coba Aplikasi

No	Penilaian	Presentase Skor
1	Operasional Sistem Aplikasi	87,5%
2	User Interface Aplikasi	87,5%
Rata-Rata Presentase		87,5%

KESIMPULAN

Konsep OOP atau *Objected Oriented Programming* pada java ini, memiliki banyak sekali kegunaan untuk menciptakan sebuah program dari masalah kehidupan sehari-hari ini. Tidak salah memang bahwa banyak hal yang bisa diimplementasikan dari kehidupan sehari-hari dengan OOP java ini. Berdasarkan hasil uji coba pengguna, hasil kepuasan pengguna berdasarkan aspek penilaian operasional sistem dan *user interface* aplikasi sebesar 87,5% yang berarti sangat baik. Sehingga dapat disimpulkan bahwa aplikasi persuratan sederhana berbasis java ini layak untuk digunakan. Program ini, dapat diterapkan pada pengguna khususnya tempat kantor di kota-kota Indonesia. Adanya system ini dengan Konsep OOP dapat memberikan kenyamanan bagi pengguna dan membuat kinerja lebih efisien.

DAFTAR PUSTAKA

[1]. Sasmito, A. 2017. Rancangan Pembelajaran Pemrograman Java Dengan Pendekatan Object-First Untuk Mempermudah Memahami Konsep Object Oriented Programing (OOP). Jurnal Pilar Nusa Mandiri, Vol.10 No.2 Halaman 149.

[2]. Hermansyah, H., Juhriah, E., & Saragih, M. A. (2021). Perancangan Aplikasi Surat Masuk dan Surat Keluar Berbasis Java di PT Afconsult Energy Indonesia. Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI), 2(01).

[3]. Sari, G. P., Marzal, J., & Mauladi, M. (2018). Rancang Bangun Sistem Informasi Persuratan dan Disposisi Elektronik Universitas Jambi. JUSS (Jurnal Sains dan Sistem Informasi), 1(1), 20-29.

[4]. Wicaksana, A. F., & Manuputty, A. D. (2020). Perancangan Sistem Informasi Persuratan Berbasis Desktop Di Bagian Sekretariat Dinas Pekerjaan Umum Kota Semarang. Jurnal Bina Komputer, 2(2), 29-38.

[5]. Sudirman, S. (2021). Machine Learning Deteksi Jatuh Menggunakan Algoritma Human Posture Recognition. Proceeding KONIK (Konferensi Nasional Ilmu Komputer), 5, 462-466.

[6]. Sudirman, S., Zainuddin, Z., & Suyuti, A. (2021). Fall Detection in the Elderly With Android Mobile IoT Devices Using Nodemcu And Accelerometer Sensors.

[7]. Putri, G. A. A. (2005). Rancang Bangun Sistem Informasi Persuratan Dan Kearsipan Universitas Udayana Menggunakan Paradigma Pemrograman Berorientasi Objek. Teknologi Elektro, 4(2), 35-41.