KONSEP PEMROGRAMAN BERORIENTASI OBJECT (PBO) PADA DELPHI

EKA MEI RISTIANTI

165100091, 785567914

Fakultas Komputer

Ekameiristianti.student@umitra.ac.id

Abstract

Merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Program ini mempunyai kemampuan luas yang terletak pada produktifitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta bahasa pemrogramannya terstruktur dan lengkap. Fasilitas pemrograman dibagi dalam dua kelompok yaitu object dan bahasa pemrograman.

Object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat. Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman dapat disebut sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu untuk menjalankan tugas tertentu. Gabungan antara object dengan bahasa pemrograman sering disebut bahasa pemrograman berorientasi object.

OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programming yang berarti pula Pemrograman Berorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Kata Kunci : Object Oriented Program (OOP)

A. INTRODUCTION

Borland Delphi

Merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Program ini mempunyai kemampuan luas yang terletak pada produktifitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta bahasa pemrogramannya terstruktur dan lengkap. Fasilitas pemrograman dibagi dalam dua kelompok yaitu object dan bahasa pemrograman. Object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat. Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman dapat disebut sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu untuk menjalankan tugas tertentu. Gabungan antara object dengan bahasa pemrograman sering

disebut bahasa pemrograman berorientasi object.

Apa itu PBO?

OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programming yang berarti pula Pemrograman Berorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Pemrograman Berorientasi Objek (Object Oriented Programming/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan

Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu:

Attribute

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas, Atribut dapat memiliki hak akses private, public maupun protected.

Sebuah atribut yang dinyatakan sebagai private hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas

lainnya tidak dapat mengakses atribut ini secara langsung

Method

Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu Class. Dengan kata lain, method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrograman prosedural.

Pada sebuah method di dalam sebuah kelas juga memiliki izin akses seperti halnya atribut pada kelas, izin akses itu antara lain private, public dan protected yang memiliki arti sama pada izin akses atribut yang telah dibahas sebelumnya. Sebuah kelas boleh memiliki lebih dari satu method dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga kompiler atau interpreter dapat mengenali method mana yang dipanggil.

Class

Class didefinisikan sebagai sebuah blueprint(denah), atau prototipe, yang

mendefinisikan variabel-variabel dan metode-metode yang umum untuk semua objek dari n jenis tertentu (n maksudnya jumlah tertentu). Sebuah kelas menyerupai sebuah struktur yang merupakan tipe data sendiri, misalkan tipe data titik yang terdiri dari koordinat x dan y. Bahasa Java telah menggunakan sebuah kelas untuk menyatakan tipe data titik karena bahasa pemrograman Java merupakan bahasa pemrograman berorientasi objek murni sehingga tidak mengenal struktur, melainkan mengenal apa yang disebut dengan kelas.

Object

Merupakan perwujudan dari class, setiap object akan mempunyai attribute dan method yang dimiliki oleh class-nya, contohnya: amir, ahmad, yani merupakan object dari class manusia. Setiap object dapat berinteraksi dengan object lainnya meskipun berasal dari class yang berbeda.

Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun object lain dan

mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya. Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah object. Rumah, mobil, sepeda motor, meja, dan komputer merupakan contoh-contoh object yang ada di dunia nyata. Object adalah implementasi dari class. Secara sederhananya, dapat dikatakan terdiri dari properti (atribut) dan method.

Konsep-konsep dalam OOP/PBO :

Abstraction

Abstraksi merupakan kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

Encapsulation

Merupakan suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem tersebut.

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak. Hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

Inheritance

Merupakan konsep mewariskan attribute dan method yang dimiliki oleh sebuah class kepada class turunannya. Dengan konsep ini class yang dibuat cukup mendefinisikan attribute dan method yang spesifik didalamnya, sedangkan attribute dan method yang lebih umum akan didapatkan dari class yang menjadi induknya.

Mengatur polimorfisme dan enkapsulasi dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).

Polymorphism

Merupakan konsep yang memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu object agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda.

Model data berorientasi objek disebut dapat memberi fleksibilitas yang lebih, kemudahan mengubah program dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya selain itu pendekatan OOP lebih mudah dikembangkan dan dirawat.

B. CONTENT

Studi Kasus "Food Court 'Oishi Ne'"

Program OOP (*Object Oriented Program*) dengan Studi Kasus "Food Court 'Oishi Ne'"

Berikut analisa Kasusnya!

Food Court "Oishi Ne" merupakan rumah makan yang menyediakan makanan dimana masing-masing menu makanan disediakan dalam bentuk paket nama, menu paket dan harga.

Untuk menu minuman, Food Court 'Oishi Ne' menyediakan Air Mineral,

berbagai macam Juice dengan harga yang sama dan Es Teh Manis

Agar lebih dekat dengan pelanggan dan meningkatkan pelayanan, Food Court 'Oishi Ne' memberikan pelayanan tambahan yaitu delivery dan take away.

Proses Program:

1. Untuk memulai program, kasir harus menekan tombol bersih agar imputan transaksi sebelumnya hilang dan semua objek dalam program siap digunakan.
2. Pelanggan memilih paket makanan yang akan dibeli. dengan memilih salah satu paket makan nama menu dan harga paket akan tampil di masing-masing textfield.
3. pelanggan menentukan jumlah menu yang dibeli dengan tugas kasir adalah menginput jumlah beli sesuai dengan keinginan pelanggan. Dengan menekan tombol enter maka total harga yang harus dibayar oleh pelanggan akan tampil sesuai

dengan kalkulasi harga paket dan jumlah yang dibeli.

4. Sebelum transaksi selesai, pelanggan diperbolehkan untuk memilih minuman.
5. Ketika diinput jumlah minumannya, maka total bayar akan bertambah sesuai dengan hasil kalkulasi pembelian minuman. Contoh: Jika total bayar 1000 dan pelanggan membeli air mineral 2 gelas dengan total harga 1000, maka total bayar menjadi 2000.
6. Begitu juga dengan jenis pelayanannya. Untuk pelayanan delivery, Food Court 'Oishi Ne' memberikan tarif 10% dari total bayar. Sedangkan untuk take away tidak dikenakan tarif. Tarif pelayanan akan muncul otomatis disebuah TextField.
7. Sleanjutkan kasir akan menekan tombol 'Hitung' sehingga total bayar akan dijumlahkan dengan tarif pelayanannya.
8. Kasir akan menginput uang bayar yang diberikan

pelanggan, Jika uangnya melebihi dari total yang harus dibayarkan, makan akan muncul otomatis uang kembaliannya.

9. Jika transaksi selesai maka kasir dapat menekan tombol keluar, dimana pada saat menekan tombol tersebut, program mengkonfirmasi kasir dengan memberikan pertanyaan "Apakah anda yakin ingin menutup aplikasi?". Jika kasir memilih Yes maka program ditutup dan sebaliknya.

Dari analisa tersebut diatas, maka designlah program tersebut dengan menggunakan Program PBO! Pertama, buatlah class java dengan Java Class yang diberi nama 'FormFood' (atau sesuaikan dengan keinginan anda)

Kemudian tulis coding dibawah ini pada Java Class yang anda buat tadi.

```
public class FormFood {
```

```
public String hrg_mkn=0; } }
paket,makan,minum,mnm,layan;

public int jml_mkn, jml_mnm;

public double hrg_mkn,
hrg_mnm,biaya,ub,uk, ttl, total, ttlbyr;

public void setPaket ( String val )
{ paket=val; }
public String getPaket()
{ return paket; }

public void setPilihPaket ()
{ if (paket.equalsIgnoreCase("PAKET
1"))
{ makan="Sayur Asem, Tempe
Goreng, Kerupuk";
hrg_mkn=7000; }
else if
(paket.equalsIgnoreCase("PAKET 2"))
{ makan="Ayam Goreng, Sambel,
Balado Kentang";
hrg_mkn=10000; }
else if
(paket.equalsIgnoreCase("PAKET 3"))
{ makan="Balado Telur, Sayur Lodeh,
Tahu";
hrg_mkn=8500; }
else
{ makan="";

public void setMakan(String val)
{ makan=val; }
public String getMakan()
{ return makan; }

public void setHrgMkn(Double val)
{ hrg_mkn=val; }
public double getHrgMkn()
{ return hrg_mkn; }

public void setJmlMkn(Integer val)
{ jml_mkn=val; }
public int getJmlMkn()
{ return jml_mkn; }

public void setMinum (String val)
{ minum=val; }
public String getMinum()
{ return minum; }

public void setPilihMinum()
{ if ( minum.equalsIgnoreCase("ES
TEH MANIS"))
{ hrg_mnm=2500; }
else if (
minum.equalsIgnoreCase("JUS"))
{ hrg_mnm=5000; }
```



```
else if ( minum.equalsIgnoreCase("AQUA BOTOL"))
{ hrg_mnm=500; }
else { hrg_mnm=0; } }

public void setHrgMnm(Double val)
{ hrg_mnm=val; }

public double getHrgMnm()
{ return hrg_mnm; }

public void setJmlMnm(Integer val)
{ jml_mnm=val; }

public int getJmlMnm()
{ return jml_mnm; }

public void setLayan(String val)
{ layan = val; }

public String getLayan()
{ return layan; }

public void setSeleksiLayan()
{ if (layan.equalsIgnoreCase("Take A Way"))
{ biaya=0; }
else if (layan.equalsIgnoreCase("Delivery"))
{
biaya=0.1*((hrg_mkn*jml_mkn)+(hrg_mnm*jml_mnm)); }

else { biaya=0; }
}

public void setBiaya(Double val)
{ biaya=val; }

public double getBiaya()
{ return biaya; }

public void setTtl(Double val)
{ ttl=val; }

public double getTtl()
{ return (hrg_mkn*jml_mkn); }

public void setTotal(Double val)
{ total=val; }

public double getTotal()
{
return ((hrg_mkn*jml_mkn)+(hrg_mnm*jml_mnm)); }

public void setTtl_byr(Double val)
{ ttlbyr=val; }

public double getTtl_byr()
{
ttlbyr=(hrg_mkn*jml_mkn)+(hrg_mnm*jml_mnm)+biaya;
return (ttlbyr); }

public void setUb(Double val)
{ ub=val; }
```

25

```
        dispose();
    else
        return;
}

private void
cmakanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    food.setPaket(cmakan.getSelectedItem().toString());
    food.setPilihPaket();
    tmenu.setText(food.getMakan());

    thrghmenu.setText(Double.toString(food.getHrgMkn()));
    tjmlmkn.grabFocus();
}

private void
cminumActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    food.setMinum(cminum.getSelectedItem().toString());
    food.setPilihMinum();
```

```
    thrghmnm.setText(Double.toString(food.getHrgMnm()));
    tjmlmnm.grabFocus();
}

private void
tjmlmknKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:

    if
    (evt.getKeyCode()==KeyEvent.VK_ENTER)
    {
        food.setJmlMkn(Integer.parseInt(tjmlmkn.getText()));
        tttotal.setText(Double.toString(food.getTtl()));
    }
}

private void
tjmlmnmKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
```

```
        if
    (evt.getKeyCode()==KeyEvent.VK_E
    NTER)

        {
    food.setJmlMnm(Integer.parseInt(tjml
    mnm.getText()));

    tttotal.setText(Double.toString(food.get
    Total()));
        }
    }

    private void
    rbtakeActionPerformed(java.awt.event
    .ActionEvent evt) {
        // TODO add your handling code
    here:

        food.setLayan(rbtake.getText());
        food.setSeleksiLayan();

    ttake.setText(Double.toString(food.get
    Biaya()));
        tdeliv.setText("");
    }

    private void
    rbdelivActionPerformed(java.awt.even
    t.ActionEvent evt) {
        // TODO add your handling code
    here:
```

```
        food.setLayan(rbdeliv.getText());
        food.setSeleksiLayan();

    tdeliv.setText(Double.toString(food.get
    Biaya()));
        ttake.setText("");
    }

    private void
    bhitungActionPerformed(java.awt.eve
    nt.ActionEvent evt) {
        // TODO add your handling code
    here:

    tttotal.setText(Double.toString(food.get
    Ttl_byr()));
        tubay.grabFocus();
    }

    private void
    tubayKeyPressed(java.awt.event.KeyE
    vent evt) {
        // TODO add your handling code
    here:

        if
    (evt.getKeyCode()==KeyEvent.VK_E
    NTER)
        {
    food.setUb(Double.parseDouble(tubay
    .getText()));
```

```
tukem.setText(Double.toString(food.ge  
tUk()));
```

```
if(food.getUb()<food.getTtl_byr())  
    {  
JOptionPane.showMessageDialog(nul  
l,"Pembayaran          Anda  
Kurang!!","Warning",JOptionPane.O  
K_OPTION);  
        tukem.setText("");  
        tubay.setText("");}  
    else { return; }  
    }  
}
```

```
void bersih()  
{ cmakan.setSelectedItem("--PILIH-  
-");  
    cminum.setSelectedItem("--PILIH-  
-");  
    tmenu.setText("");  
    thrghmenu.setText("");  
    tjmlmkn.setText("");  
    thrghmnm.setText("");  
    tjmlmnm.setText("");  
    pilihan.clearSelection();  
    ttake.setText("");  
    tdeliv.setText("");  
    tttotal.setText("");
```

```
tubay.setText("");  
tukem.setText(""); }
```

```
void aktif()  
{ cmakan.setEnabled(true);  
    cminum.setEnabled(true);  
    tjmlmkn.setEnabled(true);  
    tjmlmnm.setEnabled(true);  
    rbtake.setEnabled(true);  
    rbdeliv.setEnabled(true);  
    tubay.setEnabled(true); }
```

```
void nonaktif()  
{ cmakan.setEnabled(false);  
    cminum.setEnabled(false);  
    tmenu.setEnabled(false);  
    thrghmenu.setEnabled(false);  
    thrghmnm.setEnabled(false);  
    tjmlmkn.setEnabled(false);  
    tjmlmnm.setEnabled(false);  
    rbtake.setEnabled(false);  
    rbdeliv.setEnabled(false);  
    ttake.setEnabled(false);  
    tdeliv.setEnabled(false);  
    tttotal.setEnabled(false);  
    tubay.setEnabled(false);  
    tukem.setEnabled(false); }
```

C. CONCLUSION

Jadi kesimpulan dari penjelasan di atas ialah bahwa Delphi Merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web.

Sedangkan OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programming yang berarti pula Pemrograman Berorientasi. Pemrograman Berorientasi Objek (Object Oriented Programming/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data.

Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan

satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu:

- Attribute
- Method
- Class
- Object

Konsep-konsep dalam OOP/PBO :

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Dan disini juga dijelaskan contoh study kasus Pemrograman Berorientasi Objek (PBO) dengan menggunakan JAVA .

D. DISCUSSION

Hasil diskusi tersebut ialah Borland Delphi Merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Program ini mempunyai kemampuan luas yang terletak pada produktifitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta bahasa pemrogramannya terstruktur dan lengkap. Object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat. Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman dapat disebut sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu untuk menjalankan tugas tertentu. Gabungan antara object dengan bahasa pemrograman sering disebut bahasa pemrograman berorientasi object.

OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programing yang berarti pula Pemrograman Beorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Pemrograman Berorientasi Objek (Object Oriented Programming/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Disini juga menyertakan contoh study kasus menggunakan JAVA Program OOP (*Object Oriented Program*) dengan Studi Kasus "Food Court 'Oishi Ne'"

yang disertai dengan proses program dan codingnya .

E. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra, "COMPREHENSIVE SET OF PROFESSIONAL FOR DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU-POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."
- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra, "IMPLEMENTATION OF PARADOX DBASE."
- [11] A. S. Putra, "IMPLEMENTATION OF TRADE SECRET CASE STUDY SAMSUNG MOBILE PHONE."
- [12] A. S. Putra, "IMPLEMENTATION PATENT FOR APPLICATION WEB BASED CASE STUDI WWW. PUBLIKLAMPUNG.COM."
- [13] A. S. Putra, "IMPLEMENTATION SYSTEM FIRST TO INVENT IN DIGITALLY INDUSTRY."
- [14] A. S. Putra, "MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0."
- [15] A. S. Putra, "PATENT AS RELEVAN SUPPORT RESEARCH."
- [16] A. S. Putra, "PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc."
- [17] A. S. Putra, "PATENT PROTECTION FOR APPLICATION INVENT."
- [18] A. S. Putra, "QUICK REPORT IN PROPERTY PROGRAMMING."
- [19] A. S. Putra, "REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK."
- [20] A. S. Putra, "REVIEW TRADEMARK PATENT FOR

- INDUSTRIAL
TECHNOLOGY BASED 4.0.”
- [21] A. S. Putra, “TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING.”
- [22] A. S. Putra, “WORKING DIRECTORY SET FOR PARADOX 7.”
- [23] A. S. Putra, “ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk.”
- [24] A. S. Putra, D. R. Aryanti, and I. Hartati, “Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya),” in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 85–97.
- [25] A. S. Putra and O. M. Febriani, “Knowledge Management Online Application in PDAM Lampung Province,” in *Prosiding International conference on Information Technology and Business (ICITB)*, 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, “Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung,” *SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data)*, vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K. Zuhri, “Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, pp. 221–232.
- [28] D. P. Sari, O. M. Febriani, and A. S. Putra, “Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya,” in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.