

REPUBLIC INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202277899, 21 Oktober 2022

Pencipta

Nama : **Irwan Alnarus Kautsar, S.Kom., M.Kom., Ph.D, M. Ruslianor
Maika, S.Hut, M.A.B dkk**

Alamat : Perumahan Pucang Indah M-02, Sidoarjo, Sidoarjo, JAWA TIMUR,
61219

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Universitas Muhammadiyah Sidoarjo**

Alamat : Jl. Mojopahit 666-B, Sidoarjo, Sidoarjo, JAWA TIMUR, 61215

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Modul Rapid Prototyping Menggunakan Metode Cause Effect
Solution (CES) Pada Platform Supportive Tool Dalam Rangka
Menunjang Model Pembelajaran Case Based Learning**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 19 Oktober 2022, di Sidoarjo

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000393640

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

Modul Rapid Prototyping Menggunakan Metode Cause Effect Solution (CES) pada Platform Supportive Tool dalam rangka Menunjang Model Pembelajaran Case Based Learning

Inovasi-inovasi pada peningkatan perekonomian, perdagangan dan transaksi keuangan merupakan salah satu unsur vital dalam hal kemandirian bangsa. Terutama pada era Revolusi Industri 4.0, merupakan era industri yang mentransformasi transaksi-transaksi keuangan konvensional ke arah digital. Untuk itu civitas akademika dalam Pendidikan Tinggi sangatlah diharapkan menjadi katalisator inovasi-inovasi pada sektor-sektor tersebut. Katalisasi inovasi tersebut secara real dapat dilakukan dengan mengkolaborasikan pembelajaran dan penelitian multidisiplin antara mahasiswa Program Studi Informatika dan mahasiswa Program Studi Perbankan Syariah dalam rangka peningkatan Inovasi Keuangan Digital berbasis akad syariah yang melatarbelakangi penciptaan ini.

Disisi lain, kebutuhan masyarakat terhadap inovasi-inovasi berupa produk-produk keuangan syariah sangatlah besar. Akan tetapi, inovasi-inovasi tersebut kebanyakan dihasilkan oleh lembaga-lembaga/startup digital di luar institusi Pendidikan Tinggi, yang tentunya target utama dari perusahaan tersebut ialah komersial (jual beli) . Hal ini berbeda dengan tujuan adanya akad syariah. Tujuan utama muamalah akad syariah adalah menjadikannya proses edukasi dan jaminan tidak adanya kerugian diantara kedua belah pihak. Untuk itu, kolaborasi penelitian dalam bentuk “Collaborative Project” pada Program Studi Informatika dan Perbankan Syariah dapat menjadi salah satu sarana bagi peningkatan mutu pendidikan tinggi untuk menghasilkan SDM yang berkualitas. Serta peningkatan kemampuan Iptek dan inovasi untuk

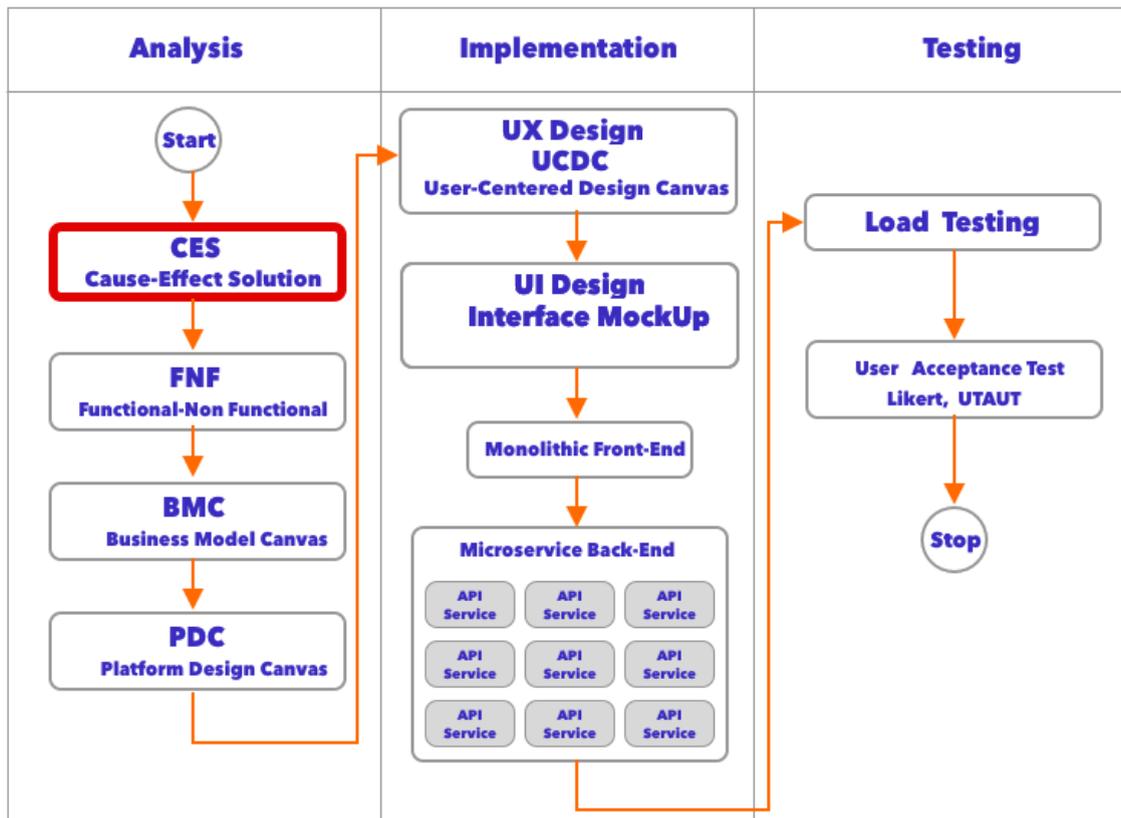
menghasilkan nilai tambah produk inovasi (Permenristekdikti No. 50 Tahun 2017).

Tujuan dari penciptaan ini adalah untuk merancang bangun sebuah kerangka kerja/framework dan platform prototyping yang nantinya akan digunakan oleh dosen dan mahasiswa Prodi Informatika dan Perbankan syariah untuk berkolaborasi menghasilkan inovasi-inovasi keuangan digital berdasarkan konsep-konsep akad syariah. Sebagai contoh: akad Mudharabah, Ijarah dan Murabahah.

Termasuk dalam penciptaan ini adalah 1). Mengembangkan Kerangka kerja/Framework dan Platform yang dapat menjadi alat pendukung (Supportive Tools) untuk proses pembelajaran dan/atau prototyping yang lebih adaptable pada mahasiswa Informatika dan Perbankan Syariah. Sehingga mahasiswa Informatika dan Perbankan Syariah dapat berkolaborasi untuk merancang bangun teknologi keuangan digital berbasis akad syariah. 2). Kerangka kerja/Framework yang dikembangkan dapat menjadi model pembelajaran multi disiplin pada prodi lainnya. 3). Platform yang dikembangkan sebagai alat bantu dosen pembina dan stakeholder lain untuk memonitor dan mempresentasikan hasil kinerja dari kelompok-kelompok studi yang dibentuk pada kolaborasi lintas program studi tersebut.

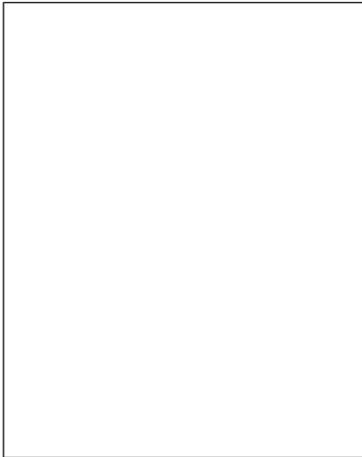
Framework yang dikembangkan adalah: (1) Framework Cause-Effect-Solution (CES); Tahapan implementasi framework prototyping yang telah dikembangkan ada pada Gambar 1. Penggunaan CES Framework tersebut dengan mendefinisikan permasalahan yang terjadi di sistem eksisting. berbeda dengan kerangka kerja Cause-Effect yang umumnya digambarkan menggunakan diagram tulang ikan (Fishbone). Cause-Effect-Solution merupakan analisa permasalahan sistem eksisting menggunakan canvas/isian seperti Business

Model Canvas (BMC), Platform Design Canvas (PDC) dan User-Centered Design Canvas. Untuk menggunakan framework CES, mahasiswa diminta mengisi 3 kolom yaitu kolom Cause, Effect dan Solution. Framework CES ini merupakan tahapan awal sebelum mendefinisikan Fitur-fitur Functional dan Non Functional pada platform yang akan dikembangkan. Tujuan dari implementasi framework CES adalah membantu mahasiswa dengan lebih mudah mendefinisikan solusi dari masalah eksisting sebagai bagian dari implementasi prototyping dan Case Based Learning.



Gambar 1. Tahapan Rapid Prototyping

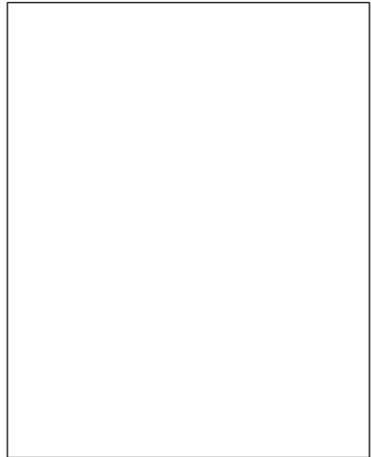
Cause

An empty rectangular box with a thin black border, intended for writing the cause of a problem.

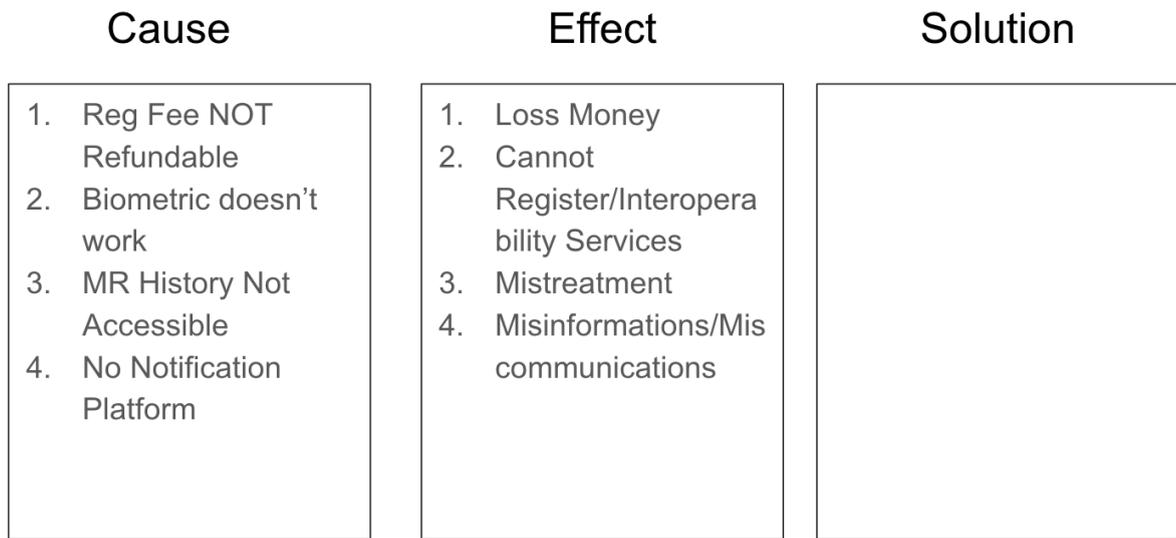
Effect

An empty rectangular box with a thin black border, intended for writing the effect of a problem.

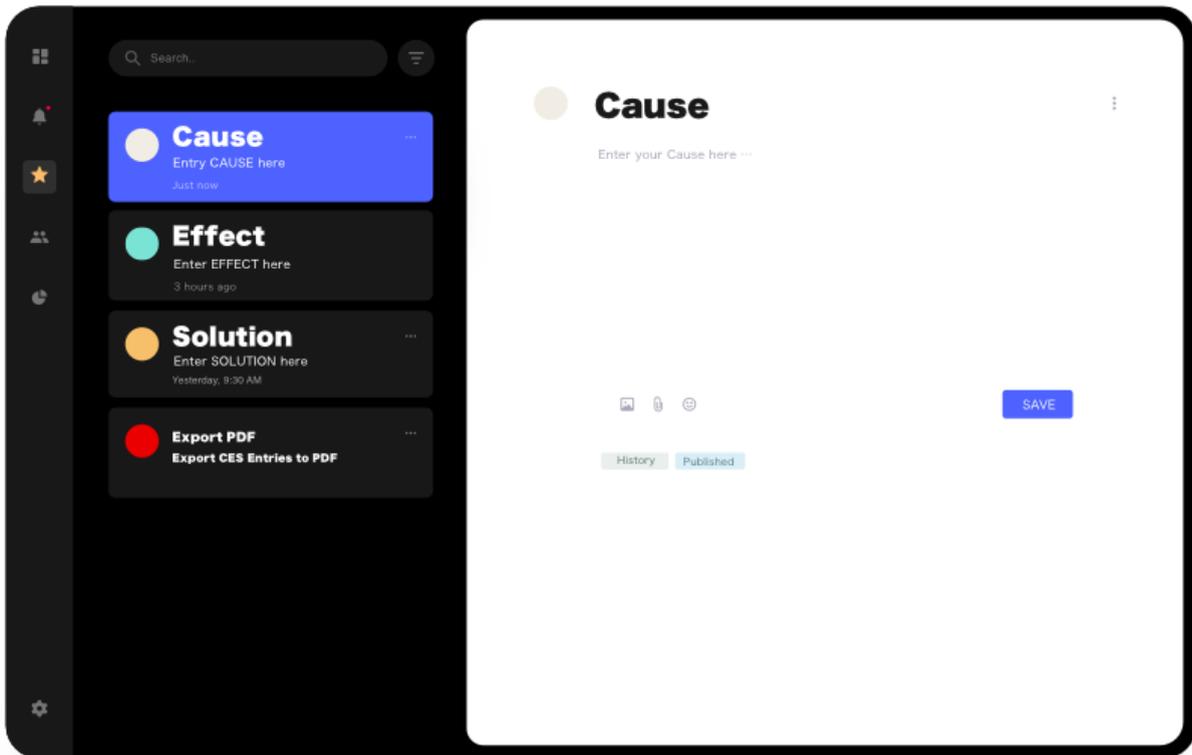
Solution

An empty rectangular box with a thin black border, intended for writing the solution to a problem.

Gambar 2. Format Canvas CES Framework



Gambar 3. Contoh inputan Canvas CES Framework



Gambar 4. User Interface (UI) Canvas CES Framework

```

from flask import Flask, jsonify
from flask_restful import Api, Resource, reqparse
from peewee import *

db = "ces.db"
database = SqliteDatabase(db)

class BaseModel(Model):
    class Meta:
        database=database

class ces(BaseModel):
    idces = AutoField(primary_key=True)
    kodekelompok = TextField()
    type = IntegerField() # 1 for cause, 2 for efect, 3 for solution
    name = CharField()

def create_tables():
    with database:
        database.create_tables([ces])

```

Gambar 5. Inisiasi Database Module CES Framework

```
app = Flask(__name__)
api = Api(app)

class resourceCes(Resource):
    def get(self): ==
    def post(self): ==
    def put(self): ==
    def delete(self): ==

api.add_resource(resourceCes, '/')

if __name__ == '__main__':
    create_tables()
    app.run(port=5001, debug=True)
```

Gambar 6. Inisiasi Web Service Module CES Framework

```

def get(self):
    parser = reqparse.RequestParser()
    parser.add_argument("kodekelompok", location="args")
    parser.add_argument("idces", location="args")
    args = parser.parse_args()

    if args['kodekelompok'] and args['idces']:
        return jsonify({"data":None, "message":"Too Many Parameters"})
    elif args['kodekelompok']:
        data_ces = list(ces.select().where(ces.kodekelompok == args['kodekelompok']).dicts())
    elif args['idces']:
        data_ces = list(ces.select().where(ces.idces == args['idces']).dicts())
    else:
        data_ces = list(ces.select().dicts())
    return jsonify({"data":data_ces, "message":"success"})

```

Gambar 7. Implementasi Service CES Framework: GET

```
def post(self):
    parser = reqparse.RequestParser()
    parser.add_argument("kodekelompok", location="json")
    parser.add_argument("type", location="json")
    parser.add_argument("name", location="json")
    args = parser.parse_args()

    ces.create(kodekelompok=args['kodekelompok'], type=args['type'], name=args['name'])
    return jsonify({"data":None, "message":"create CES success"})
```

Gambar 8. Implementasi Service CES Framework: POST

```
def put(self):
    parser = reqparse.RequestParser()
    parser.add_argument('idces', location='json')
    parser.add_argument('name', location='json')

    args = parser.parse_args()

    # cek if ces exists
    cek = ces.select().where(ces.idces == args['idces'])
    if cek.exists():
        update_ces = ces.update(name=args['name']).where(ces.idces==args['idces'])
        update_ces.execute()
        return jsonify({"data":None, "message":"Update CES success"})
    else:
        return jsonify({"data":None, "message":"Update CES Failed. CES not Found"})
```

Gambar 9. Implementasi Service CES Framework: PUT

```
def delete(self):
    parser = reqparse.RequestParser()
    parser.add_argument('idces', location="args")
    args = parser.parse_args()
    # cek if ces exists
    cek = ces.select().where(ces.idces == args['idces'])
    if cek.exists():
        delete_ces = ces.delete().where(ces.idces == args['idces'])
        delete_ces.execute()
        return jsonify({"data":None, "message":"Delete CES success"})
    else:
        return jsonify({"data":None, "message":"Delete CES Failed. CES not Found"})
```

Gambar 10. Implementasi Service CES Framework: DELETE

```

class TaskCES(TaskSet):
    @task
    def getCESbykodekelompok(self):
        kodekelompok = random.randint(1,1000)
        self.client.get(f"?kodekelompok={kodekelompok}")

    @task
    def getCESbyidfnf(self):
        idces = random.randint(1,1000)
        self.client.get(f"?idces={idces}")

    @task
    def postCES(self):
        kodekelompok = random.randint(1,1000)
        tipe = random.randint(1,9)
        ranuuid = str(uuid.uuid4()).hex
        self.client.post('/', json={"kodekelompok":kodekelompok, "type":tipe, "name":"ranuuid"})

    @task
    def putCES(self):
        ranuuid = str(uuid.uuid4()).hex
        idfnf = random.randint(1,1000)
        self.client.put("/", json={"idfnf":idfnf, "name":ranuuid})

    @task
    def deleteCES(self):
        idces = random.randint(1,1000)
        self.client.get(f"?idces={idces}")

```

Gambar 11. Implementasi Service CES Framework: Load Initiation

CES Framework Load Test Report

During: 9/18/2022, 11:16:04 PM - 9/18/2022, 11:18:00 PM

Target Host: http://127.0.0.1:5001

Script: cesLoad.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/	1008	0	23	6	155	55	8.7	0.0
PUT	/	1008	0	20	4	203	69	8.7	0.0
GET	/?ldces=1	3	0	19	6	44	148	0.0	0.0
GET	/?ldces=10	2	0	6	6	7	149	0.0	0.0
GET	/?ldces=100	2	0	33	17	48	150	0.0	0.0
GET	/?ldces=101	3	0	10	6	18	150	0.0	0.0
GET	/?ldces=102	1	0	5	5	5	150	0.0	0.0
GET	/?ldces=103	2	0	49	8	91	150	0.0	0.0
GET	/?ldces=104	1	0	6	6	6	149	0.0	0.0
GET	/?ldces=105	3	0	41	5	104	150	0.0	0.0
GET	/?ldces=106	2	0	7	5	8	150	0.0	0.0
GET	/?ldces=107	2	0	48	7	89	150	0.0	0.0
GET	/?ldces=108	1	0	41	41	41	150	0.0	0.0
GET	/?ldces=109	3	0	46	6	85	150	0.0	0.0
GET	/?ldces=11	5	0	20	5	59	149	0.0	0.0
GET	/?ldces=110	2	0	7	6	8	150	0.0	0.0
GET	/?ldces=111	1	0	44	44	44	150	0.0	0.0
GET	/?ldces=113	3	0	20	6	48	114	0.0	0.0

Gambar 12. CES Framework Load Test Report

Modul Rapid Prototyping Menggunakan Metode Cause Effect Solution

by Irwan Kautsar

Submission date: 06-Aug-2023 02:59PM (UTC+0700)

Submission ID: 2141926793

File name: CC1-CES.pdf (776.99K)

Word count: 566

Character count: 3959

Modul Rapid Prototyping Menggunakan Metode Cause Effect Solution (CES) pada Platform Supportive Tool dalam rangka Menunjang Model Pembelajaran Case Based Learning

Inovasi-inovasi pada peningkatan perekonomian, perdagangan dan transaksi keuangan merupakan salah satu unsur vital dalam hal kemandirian bangsa. Terutama pada era ⁴ Revolusi Industri 4.0, merupakan era industri yang mentransformasi transaksi-transaksi keuangan konvensional ke arah digital. Untuk itu civitas akademika dalam Pendidikan Tinggi sangatlah diharapkan menjadi katalisator inovasi-inovasi pada sektor-sektor tersebut. Katalisasi inovasi tersebut secara real dapat dilakukan dengan mengkolaborasikan pembelajaran dan penelitian multidisiplin ² antara mahasiswa Program Studi Informatika dan mahasiswa Program Studi Perbankan Syariah dalam rangka peningkatan Inovasi Keuangan Digital berbasis akad syariah yang melatarbelakangi penciptaan ini.

Disisi lain, kebutuhan masyarakat terhadap inovasi-inovasi berupa produk-produk keuangan syariah sangatlah besar. Akan tetapi, inovasi-inovasi tersebut kebanyakan dihasilkan oleh lembaga-lembaga/startup digital di luar institusi Pendidikan Tinggi, yang tentunya target utama dari perusahaan tersebut ialah komersial (jual beli) . Hal ini berbeda dengan tujuan adanya akad syariah. Tujuan utama muamalah akad syariah adalah menjadikannya proses edukasi dan jaminan tidak adanya kerugian diantara kedua belah pihak. Untuk itu, kolaborasi penelitian dalam bentuk “Collaborative Project” pada Program Studi Informatika dan Perbankan Syariah dapat menjadi salah satu sarana bagi peningkatan ¹ mutu pendidikan tinggi untuk menghasilkan SDM yang berkualitas. Serta peningkatan kemampuan lptek dan inovasi untuk

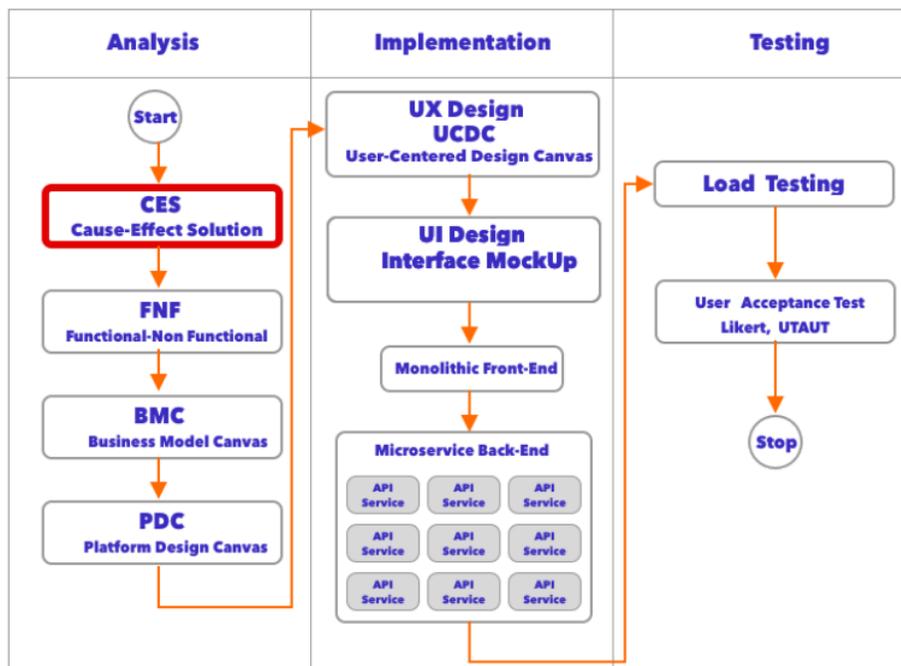
menghasilkan nilai tambah produk inovasi (Permenristekdikti No. 50 Tahun 2017).

Tujuan dari penciptaan ini adalah untuk merancang bangun sebuah kerangka kerja/framework dan platform prototyping yang nantinya akan digunakan oleh dosen dan mahasiswa Prodi Informatika dan Perbankan syariah untuk berkolaborasi menghasilkan inovasi-inovasi keuangan digital berdasarkan konsep-konsep akad syariah. Sebagai contoh: akad Mudharabah, Ijarah dan Murabahah.

Termasuk dalam penciptaan ini adalah 1). Mengembangkan Kerangka kerja/Framework dan Platform yang dapat menjadi alat pendukung (Supportive Tools) untuk proses pembelajaran dan/atau prototyping yang lebih adaptable pada mahasiswa Informatika dan Perbankan Syariah. Sehingga mahasiswa Informatika dan Perbankan Syariah dapat berkolaborasi untuk merancang bangun teknologi keuangan digital berbasis akad syariah. 2). Kerangka kerja/Framework yang dikembangkan dapat menjadi model pembelajaran multi disiplin pada prodi lainnya. 3). Platform yang dikembangkan sebagai alat bantu dosen pembina dan stakeholder lain untuk memonitor dan mempresentasikan hasil kinerja dari kelompok-kelompok studi yang dibentuk pada kolaborasi lintas program studi tersebut.

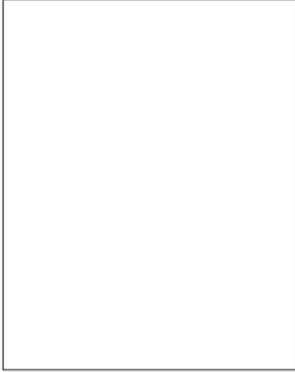
Framework yang dikembangkan adalah: (1) Framework Cause-Effect-Solution (CES); Tahapan implementasi framework prototyping yang telah dikembangkan ada pada Gambar 1. Penggunaan CES Framework tersebut dengan mendefinisikan permasalahan yang terjadi di sistem eksisting. berbeda dengan kerangka kerja Cause-Effect yang umumnya digambarkan menggunakan diagram tulang ikan (Fishbone). Cause-Effect-Solution merupakan analisa permasalahan sistem eksisting menggunakan canvas/isian seperti Business

Model Canvas (BMC), Platform Design Canvas (PDC) dan User-Centered Design Canvas. Untuk menggunakan framework CES, mahasiswa diminta mengisi 3 kolom yaitu kolom Cause, Effect dan Solution. Framework CES ini merupakan tahapan awal sebelum mendefinisikan Fitur-fitur Functional dan Non Functional pada platform yang akan dikembangkan. Tujuan dari implementasi framework CES adalah membantu mahasiswa dengan lebih mudah mendefinisikan solusi dari masalah eksisting sebagai bagian dari implementasi prototyping dan Case Based Learning.



Gambar 1. Tahapan Rapid Prototyping

Cause



Effect



Solution



Gambar 2. Format Canvas CES Framework

Cause

1. Reg Fee NOT Refundable
2. Biometric doesn't work
3. MR History Not Accessible
4. No Notification Platform

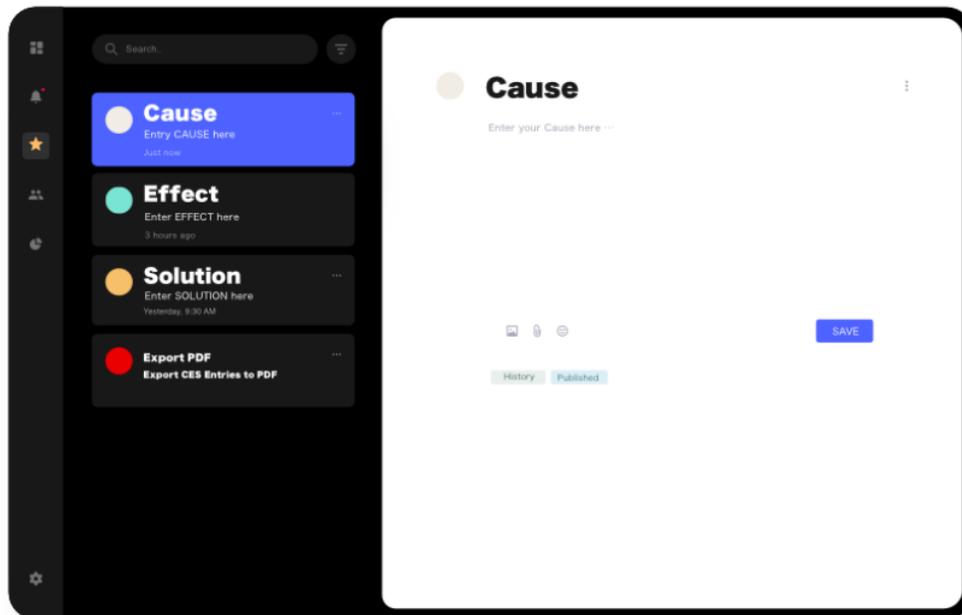
Effect

1. Loss Money
2. Cannot Register/Interoperability Services
3. Mistreatment
4. Misinformations/Miscommunications

Solution



Gambar 3. Contoh inputan Canvas CES Framework



Gambar 4. User Interface (UI) Canvas CES Framework

```
from flask import Flask, jsonify
from flask_restful import Api, Resource, reqparse
from peewee import *

db = "ces.db"
database = SqliteDatabase(db)

class BaseModel(Model):
    class Meta:
        database=database

class ces(BaseModel):
    idces = AutoField(primary_key=True)
    kodekelompok = TextField()
    type = IntegerField() # 1 for cause, 2 for efect, 3 for solution
    name = CharField()

def create_tables():
    with database:
        database.create_tables([ces])
```

Gambar 5. Inisiasi Database Module CES Framework

```
app = Flask(__name__)
api = Api(app)

class resourceCes(Resource):
    def get(self):
    def post(self):
    def put(self):
    def delete(self):

api.add_resource(resourceCes, '/')

if __name__ == '__main__':
    create_tables()
    app.run(port=5001, debug=True)
```

Gambar 6. Inisiasi Web Service Module CES Framework

```
def get(self):
    parser = reqparse.RequestParser()
    parser.add_argument("kodekelompok", location="args")
    parser.add_argument("idces", location="args")
    args = parser.parse_args()

    if args['kodekelompok'] and args['idces']:
        return jsonify({"data":None, "message":"Too Many Parameters"})
    elif args['kodekelompok']:
        data_ces = list(ces.select().where(ces.kodekelompok == args['kodekelompok']).dicts())
    elif args['idces']:
        data_ces = list(ces.select().where(ces.idces == args['idces']).dicts())
    else:
        data_ces = list(ces.select().dicts())
    return jsonify({"data":data_ces, "message":"success"})
```

Gambar 7. Implementasi Service CES Framework: GET

```
def post(self):
    parser = reqparse.RequestParser()
    parser.add_argument("kodekelompok", location="json")
    parser.add_argument("type", location="json")
    parser.add_argument("name", location="json")
    args = parser.parse_args()

    ces.create(kodekelompok=args['kodekelompok'], type=args['type'], name=args['name'])
    return jsonify({"data":None, "message":"create CES success"})
```

Gambar 8. Implementasi Service CES Framework: POST

```
def put(self):
    parser = reqparse.RequestParser()
    parser.add_argument('idces', location='json')
    parser.add_argument('name', location='json')

    args = parser.parse_args()

    # cek if ces exists
    cek = ces.select().where(ces.idces == args['idces'])
    if cek.exists():
        update_ces = ces.update(name=args['name']).where(ces.idces==args['idces'])
        update_ces.execute()
        return jsonify({"data":None, "message":"Update CES success"})
    else:
        return jsonify({"data":None, "message":"Update CES Failed. CES not Found"})
```

Gambar 9. Implementasi Service CES Framework: PUT

```
def delete(self):
    parser = reqparse.RequestParser()
    parser.add_argument('idces', location="args")
    args = parser.parse_args()
    # cek if ces exists
    cek = ces.select().where(ces.idces == args['idces'])
    if cek.exists():
        delete_ces = ces.delete().where(ces.idces == args['idces'])
        delete_ces.execute()
        return jsonify({"data":None, "message":"Delete CES success"})
    else:
        return jsonify({"data":None, "message":"Delete CES Failed. CES not Found"})
```

Gambar 10. Implementasi Service CES Framework: DELETE

```
class TaskCES(TaskSet):
    @task
    def getCESbykodekelompok(self):
        kodekelompok = random.randint(1,1000)
        self.client.get(f"/?kodekelompok={kodekelompok}")

    @task
    def getCESbyidfnf(self):
        idces = random.randint(1,1000)
        self.client.get(f"/?idces={idces}")

    @task
    def postCES(self):
        kodekelompok = random.randint(1,1000)
        tipe = random.randint(1,9)
        ranuuid = str(uuid.uuid4()).hex
        self.client.post('/', json={"kodekelompok":kodekelompok, "type":tipe, "name":"ranuuid"})

    @task
    def putCES(self):
        ranuuid = str(uuid.uuid4()).hex
        idfnf = random.randint(1,1000)
        self.client.put("/", json={"idfnf":idfnf, "name":ranuuid})

    @task
    def deleteCES(self):
        idces = random.randint(1,1000)
        self.client.get(f"/?idces={idces}")
```

Gambar 11. Implementasi Service CES Framework: Load Initiation

CES Framework Load Test Report

During: 9/18/2022, 11:16:04 PM - 9/18/2022, 11:18:00 PM

Target Host: http://127.0.0.1:5001

Script: cesLoad.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/	1008	0	23	6	155	55	8.7	0.0
PUT	/	1008	0	20	4	203	69	8.7	0.0
GET	/?ldces=1	3	0	19	6	44	148	0.0	0.0
GET	/?ldces=10	2	0	6	6	7	149	0.0	0.0
GET	/?ldces=100	2	0	33	17	48	150	0.0	0.0
GET	/?ldces=101	3	0	10	6	18	150	0.0	0.0
GET	/?ldces=102	1	0	5	5	5	150	0.0	0.0
GET	/?ldces=103	2	0	49	8	91	150	0.0	0.0
GET	/?ldces=104	1	0	6	6	6	149	0.0	0.0
GET	/?ldces=105	3	0	41	5	104	150	0.0	0.0
GET	/?ldces=106	2	0	7	5	8	150	0.0	0.0
GET	/?ldces=107	2	0	48	7	89	150	0.0	0.0
GET	/?ldces=108	1	0	41	41	41	150	0.0	0.0
GET	/?ldces=109	3	0	46	6	85	150	0.0	0.0
GET	/?ldces=11	5	0	20	5	59	149	0.0	0.0
GET	/?ldces=110	2	0	7	6	8	150	0.0	0.0
GET	/?ldces=111	1	0	44	44	44	150	0.0	0.0
GET	/?ldces=113	3	0	20	6	48	114	0.0	0.0

Gambar 12. CES Framework Load Test Report

Modul Rapid Prototyping Menggunakan Metode Cause Effect Solution

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

2%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

docplayer.info

Internet Source

3%

2

Merina Matheos, Laya M. Rares, J. S. M. Saerang. "PERBANDINGAN ANGKA KEJADIAN MIOPIA ANTARA MAHASISWA INFORMASIKA DAN ILMU KELAUTAN UNIVERSITAS SAM RATULANGI MANADO", e-CliniC, 2015

Publication

2%

3

journal.upy.ac.id

Internet Source

1%

4

onlinelearning.binus.ac.id

Internet Source

1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On