

DAFTAR PUSTAKA

- Afani, G. W., Auliasari, K., and Prasetya, R. P., 2020, Penerapan Metode Simple Multi Attribute Rating Technique Untuk Penentuan Penerima Kredit Koperasi. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 4 (1): 102–9. DOI: 10.36040/jati.v4i1.2313.
- Aprianti, W., and Maliha, U., 2016, Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati. *Jurnal Sains Dan Informatika*, 2 (1): 21–28.
- Arbian, D., 2017, Sistem Pendukung Keputusan (SPK) Pemberian Beasiswa Berbasis TOPSIS (Studi Kasus Yayasan Pendidikan Al-Hikmah Bululawang Malang). *Jurnal Ilmiah Teknologi Informasi Asia*, 11 (1): 29. DOI: 10.32815/jitika.v11i1.40.
- Harahap, M., and Muliani, A., 2019, Aplikasi Sistem Pakar Bagi Pengidap Kleptomania Menggunakan Visual Basic 2008. *Jurnal Penelitian Teknik Informatika*, 1 (2): 111–17.
- Hasbiyalloh, M., and Jakaria, D. A., 2018, Aplikasi Penjualan Barang Perlengkapan Handphone Di Zildan Cell Singaparna Kabupaten Tasikmalaya. *Jumantaka*, 1 (1): 61–70.
- Irawan, M. D., and Nasution, M. K. I., 2018, Rancang Bangun Sistem Pakar Mendiagnosa Penyakit Tanaman Kelapa Sawit Menggunakan Metode Bayes Berbasis Android (Studi Kasus : Perkebunan PTPN 4 Air Batu). *Jurnal Teknologi Informasi*, 2 (1): 15. DOI: 10.36294/jurti.v2i1.403.
- Irawan, M. D., and Simargolang, S. A., 2018, Implementasi E-Arsip Pada Program Studi Teknik Informatika. *Jurnal Teknologi Informasi*, 2 (1): 67. DOI: 10.36294/jurti.v2i1.411.
- Irianto, I., 2017, Pemilihan Perusahaan Jasa Pengiriman Barang Terbaik Menggunakan Metode Topsis. *Jurnal Teknologi Informasi*, 1 (1): 74. DOI: 10.36294/jurti.v1i1.46.

- Josi, A., 2017, Penerapan Metode Prototyping Dalam Membangun Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang). *Jti*, 9 (1): 50–57.
- Lumingkewas David, 2019, Peta Aset Universitas Sam Ratulangi Berbasis Web. *Jurnal Teknik Informatika*, 14 (1): 53–62. DOI: 10.35793/jti.14.1.2019.23837.
- Muhazir, A., Fakhriza, M., and Sutejo, E., 2017, Implementasi Metode Sequential Dalam Pencarian Pendistribusian Barang Pada Cargo Integration Sistem. *Jurnal Dan Penelitian Teknik Informatika*, 2 (2): 24–30.
- Neyfa, B. C., and Tamara, D., 2016, Special Meeting of Council. *British Medical Journal*, 1 (6001): 107–9. DOI: 10.1136/bmj.1.6001.107.
- Pasaribu, J. S., 2017, Penerapan Framework Yii Pada Pembangunan Sistem Ppdb Smp Bppi Baleendah Kabupaten Bandung. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 3 (2): 154–63. DOI: 10.33197/jitter.vol3.iss2.2017.132.
- Pasaribu, R., and Rahayu, D., 2017, Perancangan Aplikasi Lowongan Kerja Berbasis Web Dengan Menggunakan Metode Waterfall. *Seminar Nasional Teknologi Informatika, "The Future of Computer Vision,"*, 75–80.
- Permana, S. D. H., 2015, Sistem Penunjang Keputusan Pemilihan Sekolah Menengah Kejuruan Teknik Komputer Dan Jaringan Yang Terfavorit Dengan Menggunakan Multi-Criteria Decision Making. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 2 (1): 11. DOI: 10.25126/jtiik.201521123.
- Priyanti, D., 2013, Sistem Informasi Data Penduduk Pada Desa Bogoharjo Kecamatan Ngadirojo Kabupaten Pacitan. *IJNS - Indonesian Journal on Networking and Security*, 2 (4): 56.
- Ridwan, M., Parlina, I., and Satria, H., 2017, Sistem Pendukung Keputusan Dalam Merekomendasikan Smartphone Untuk Kalangan Pemula Dengan Metode TOPSIS. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, . DOI: 10.31227/osf.io/c6bdj.
- Samsudin, S., Irawan, M. D., and Harahap, A. H., 2019, Mobile App Education Gangguan Pencernaan Manusia Berbasis Multimedia Menggunakan Adobe Animate Cc. *Jurnal Teknologi Informasi*, 3 (2): 141. DOI: 10.36294/jurti.v3i2.1009.


- Sari, F., 2018, *Metode Dalam Pengambilan Keputusan*. sleman: deepublish.
- Suendri, 2018, Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer Dan Informatika*, 3 (1): 1–9.
- Sulianta, F., 2017, *Teknik Perancangan Arsitektur Sistem Informasi*. Yogyakarta: ANDI OFFSET.
- Widyatama &Suprpty, 2018, Bab II Landasan Teori. *Journal of Chemical Information and Modeling*, 53 (9): 1689–99.
- www.mysql.com, 2022. *Logo MySQL*. Dari: <https://www.mysql.com>.
- www.php.net, 2022. *Logo PHP*. Dari: <https://www.php.net>.



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

LAMPIRAN

Lampiran 1. Surat Izin Riset



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Willem Iskandar Pasar V Medan Estate 20371
Telp. (061) 6615683-6622925 Fax. 6615683

Nomor : B.1019/ST.I/ST.V.2/TL.00/11/2021
Lampiran : -
Hal : Izin Riset

11 November 2021

Yth. Bapak/Ibu Kepala Kantor Kepala Desa Bangun Raya

Assalamualaikum Wr. Wb.

Dengan Hormat, diberitahukan bahwa untuk mencapai gelar Sarjana Strata Satu (S1) bagi Mahasiswa Fakultas Sains dan Teknologi adalah menyusun Skripsi (Karya Ilmiah), kami tugaskan mahasiswa:


Nama : Apni Rahmadani Tanjung
NIM : 0702173166
Tempat/Tanggal Lahir : Sibuhuan, 01 Januari 1999
Program Studi : Sistem Informasi
Semester : IX (Sembilan)
Alamat : Link.VI Psr. Sibuhuan, Kab. Padang Lawas Kelurahan Psr, SIBUHUAN Kecamatan BARUMUN

untuk hal dimaksud kami mohon memberikan Izin dan bantuannya terhadap pelaksanaan Riset di Jl. Lintas Sibuhuan Binanga KM 1,5 Bangun Raya Kec. Barumun Kab. Padang Lawas Sumatera Utara, guna memperoleh informasi/keterangan dan data-data yang berhubungan dengan Skripsi (Karya Ilmiah) yang berjudul:

Implementasi Metode Waspas dan Topsis Dalam Penentuan Penerima Bantuan Pembangunan Jamban Pada Kabupaten Padang Lawas

Demikian kami sampaikan, atas bantuan dan kerjasamanya diucapkan terima kasih.


Medan, 11 November 2021
a.n. DEKAN
Wakil Dekan Bidang Akademik dan Kelembagaan


Digitally Signed

Dr. Abdul Halim Daulay, ST., M.Si
NIP. 198111062005011003

Tembusan:
- Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan

Lampiran 2. Surat Balasan Izin Riset

 **PEMERINTAH KABUPATEN PADANG LAWAS
KECAMATAN BARUMUN
DESA BANGUN RAYA**

Nomor : /2021 Sibuhuan, 16 Oktober 2021
Lamp :-
Perihal : Riset/Pengumpulan Data

Kepada Yth.
**Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara**
Di-
Medan

Assalamu'alaikum Wr.Wb

Sehubungan dengan surat saudara dengan Nomor: B.1019/ST.I/ST.V.2/TL.00/11.2021 tanggal 11 November 2021, tentang Izin Riset di Kantor Kepala Desa Bangun Raya guna untuk menyusun Skripsi Mahasiswa Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara dengan judul **"Implementasi Metode Waspas dan Topsis Dalam Penentuan Penerima Bantuan Pembangunan Jamban Pada Kabupaten Padang Lawas"**.

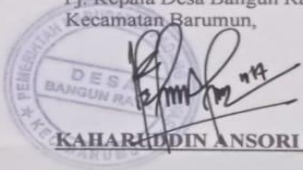
Maka dengan ini kami sampaikan bahwa Mahasiswa Universitas Islam Negeri Sumatera Utara Fakultas Sains dan Teknologi berikut ini:

Nama : Apni Rahmadani Tanjung
NIM : 0702173166
Sem : IX (Sembilan)
Prodi : Sistem Informasi

Bahwasanya dapat diizinkan untuk pengambilan data riset di Kantor Kepala Desa Bangun Raya.

Demikian kami sampikan atas perhatian kami ucapkan terimakasih.

Dikeluarkan di : Bangun Raya
Pada tanggal : 16 Oktober 2021
Pj. Kepala Desa Bangun Raya
Kecamatan Barumon,


KAHARUDDIN ANSORI NST

Lampiran 3. Tempat Observasi



Lampiran 4. Form Hasil Wawancara

FORM HASIL WAWANCARA

Tanggal Wawancara : 16 Oktober 2021

Lokasi Wawancara : Kantor Kepala Desa Bangun Raya

Narasumber : Kaharuddin Ansori Nasution

Jabatan Narasumber : Kepala Desa Bangun Raya

No.	Pertanyaan	Jawaban
1.	Siapakah nama Bapak	Kaharuddin Ansori Nasution
2.	Apakah benar ada bantuan dalam bentuk pembangunan jamban	Iya benar.
3.	Apa tujuan diberikannya bantuan jamban	Tujuannya yaitu untuk memberikan bantuan dana kesehatan kepada masyarakat kurang mampu dan mewujudkan desa <i>Open Defacation Free</i> (ODF) atau bebas buang air besar sembarangan sehingga tercipta lingkungan sehat dan bersih.
4.	Bagaimana proses penentuan penerima bantuan pembangunan jamban yang sedang berjalan	proses penentuan penerima bantuan pembangunan jamban yaitu : a. Pengumpulan data masyarakat berdasarkan kriteria yang ditentukan. b. Penyeleksian data masyarakat yang berhak menerima bantuan. c. Melakukan pembangunan jamban
5.	Kriteria apa saja dalam penentuan penerima bantuan pembangunan jamban	a. Belum memiliki jamban b. Kondisi fisik bangunan c. Jumlah penghuni rumah. d. Status sosial ekonomi e. Ketersediaan lahan

6.	Apa masalah yang terjadi saat menentukan penerima bantuan pembangunan jamban	Masalah yang terjadi yaitu banyaknya data yang harus diseleksi dengan perhitungan yang manual sehingga memungkinkan terjadinya kesalahan dalam menentukan penerima bantuan pembangunan jamban
7.	Apakah menurut bapak perlu adanya membangun sistem pendukung keputusan dalam penentuan bantuan pembangunan jamban	Iya, karena menurut saya dengan adanya sistem pendukung keputusan ini dapat mempermudah dalam penentuan bantuan pembangunan jamban.

Sibuhuan, 16 Oktober 2021

Narasumber



Khatruddin Ansori Nasution

Lampiran 5. *Black-box* Testing

Formulir *Black-box* Testing

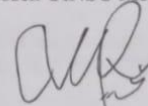
Tanggal Pengujian : 22 Agustus 2022
Nama Aplikasi : Implementasi Metode SMART dan TOPSIS dalam
Penentuan Penerima Bantuan Pembangunan Jamban pada
Desa Bangun Raya
Penguji : Ali Ikhwan, S.Kom., M.Kom
Jabatan : Dosen UINSU Medan

No.	Deskripsi	Prosedur Pengujian	Hal yang Diharapkan	Hasil
1.	Pengujian <i>Form Login</i>	Mengisi <i>username</i> dan <i>password</i> yang salah	Muncul pesan " <i>Login Gagal, Periksa Kembali Username dan Password</i> "	Sesuai
		Mengisi <i>username</i> dan <i>password</i> yang sesuai untuk dengan memasukkan (<i>username</i> = admin, <i>password</i> = 123)	Sistem akan menampilkan menu <i>dashboard</i>	Sesuai
2.	Pengujian <i>Form Alternatif</i>	Memilih menu alternatif	Sistem akan menampilkan menu alternatif	Sesuai
		Memilih button " <i>Tambah Data</i> "	Sistem akan menampilkan <i>form</i> tambah data	Sesuai
		Memilih button " <i>Edit</i> " pada salah satu data alternatif	Sistem akan menampilkan <i>form</i> edit data	Sesuai

No.	Deskripsi	Prosedur Pengujian	Hal yang Diharapkan	Hasil
		Memilih button "Hapus" pada salah satu alternatif	Sistem akan menghapus salah satu alternatif yang dipilih	Sesuai
3.	Pengujian Form Kriteria	Memilih menu kriteria	Sistem akan menampilkan menu kriteria	Sesuai
		Memilih button "Tambah Data"	Sistem akan menampilkan form tambah data	Sesuai
		Memilih button "Edit" pada salah satu kriteria	Sistem akan menampilkan form edit data	Sesuai
		Memilih button "Hapus" pada salah satu kriteria	Sistem akan menghapus salah satu kriteria yang dipilih	Sesuai
4.	Pengujian Form Sub Kriteria	Memilih menu sub kriteria	Sistem akan menampilkan menu sub kriteria	Sesuai
		Memilih button "Tambah Data"	Sistem akan menampilkan form tambah data	Sesuai
		Memilih button "Edit" pada salah satu sub kriteria	Sistem akan menampilkan form edit data	Sesuai
		Memilih button "Hapus" pada salah satu sub kriteria	Sistem akan menghapus salah satu sub kriteria yang dipilih	Sesuai

No.	Deskripsi	Prosedur Pengujian	Hal yang Diharapkan	Hasil
5.	Pengujian <i>Form</i> Penilaian	Memilih button "Edit"	Sistem akan menampilkan <i>form</i> edit data	Sesuai
		Memilih button "Simpan" pada <i>form</i> edit penilaian	Sistem akan menyimpan perubahan data yang telah diubah dan kembali ke <i>form</i> data penilaian	Sesuai
6.	Pengujian <i>Form</i> Perhitungan	Memilih menu perhitungan	Sistem akan menampilkan <i>form</i> perhitungan dengan data yang sudah diolah dengan metode SMART dan TOPSIS	Sesuai
7.	Pengujian <i>Form</i> Hasil	Memilih menu hasil akhir	Sistem akan menampilkan <i>form</i> data perhitungan dengan data yang sudah di ranking	Sesuai
		Memilih button "Cetak Data"	Sistem dapat mencetak data yang sudah di ranking dalam bentuk PDF	Sesuai

Medan, 22 Agustus 2022
Dosen UINSU Medan



Ali Ikhwan, S.Kom., M.Kom

Lampiran 6. *Sourcode*

```
<?php
namespace App;
use PDO;
include "Koneksi.php";
class Alternatif extends Koneksi{
    public function __construct()
    {
        $call = new Koneksi();
        $this->db = $call->koneksi();
    }
    public function tampil()
    {
        $sql = "SELECT * FROM
alternatif ";
        $stmt = $this->db->prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch()) {
            $data[] = $rows;
        }
        return $data;
    }
    public function hapus($val)
    {
```

```
        $sql = "SELECT * FROM alternatif
WHERE id = :id";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(":id", $val);
        $stmt->execute();
        $sql = "DELETE FROM
alternatif WHERE id = :id";
        $stmt = $this->db-
>prepare($sql);
        $stmt->bindParam(":id", $val);
        $stmt->execute()
    }
    public function update()
    {
        $sql = "SELECT * FROM
alternatif WHERE id = :id";
        $stmt = $this->db-
>prepare($sql);
        $stmt->bindParam(":id",
$_GET['id']);
        $stmt->execute();
        $sql = "UPDATE alternatif
SET nama = :nama, no_kk = :no_kk
WHERE id = :id";
        $stmt = $this->db-
>prepare($sql);
        $stmt -> bindParam(":id",
$_GET['id']);
```

```

        $stmt ->
bindParam(":nama",
$_POST['nama']);

        $stmt ->
bindParam(":no_kk",
$_POST['no_kk']);

        $stmt->execute();
    }

    public function tambah()
    {
        $sql = "INSERT INTO
alternatif VALUES (:nama,
:no_kk, :c1, :c2, :c3, :c4, :c5)";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":nama",
$_POST['kode']);

        $stmt-
>bindParam(":keterangan",
$_POST['keterangan']);

        $stmt->bindParam(":bobot",
$_POST['bobot']);

        $stmt->bindParam(":jenis",
$_POST['jenis']);

        $stmt->execute();

        header("location: kriteria.php");
    }

    public function simpan()
    {

```

```

        $sql = "INSERT INTO
alternatif VALUES (:no_kk,
:nama, " , " , " , " )";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":nama",
$_POST['nama']);

        $stmt->bindParam(":no_kk",
$_POST['no_kk']);

        $stmt->execute();
    }

    public function get_json($val)
    {
        $sql = "SELECT * FROM
alternatif WHERE id=:id limit 1";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":id", $val);

        $stmt->execute();

        //$data['data'] = array();

        $data = $stmt-
>fetchAll(PDO::FETCH_ASSOC);

        return json_encode($data);
    }
}
?>

```

2. Hasil.php

```
<?php
namespace App;
use PDO;
include "Koneksi.php";
class Hasil extends Koneksi{
    public function __construct()
    {
        $call = new Koneksi();
        $this->db = $call->koneksi();
    }
    public function countalt()
    {
        $sql = "SELECT COUNT(id)
AS count FROM alternatif";
        $stmt = $this->db-
>prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
        return $data;
    }
    public function count()
    {
        $sql = "SELECT MAX(id) AS
count FROM alternatif";
        $stmt = $this->db-
>prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
        return $data;
    }
    public function spk($b)
    {
        //Initalize
        $sql = "SELECT * FROM
alternatif WHERE id = :id";
        $stmt = $this->db-
>prepare($sql);
        $stmt->bindParam(":id", $b);
        $stmt->execute();
        //Fetch
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
    }
}
```

```

        return $data;
    }

    public function max(){
        $sql = "SELECT MAX(nilai)
AS MAX FROM hasil";
        $stmt = $this->db-
>prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
        return $data;
    }

    public function rank()
    {
        $sql = "SELECT * FROM
`hasil`, alternatif WHERE
hasil.id_alternatif=alternatif.id
ORDER BY nilai DESC LIMIT 1";

        $stmt = $this->db-
>prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
    }

```

```

    }
    return $data;
}

    public function min(){
        $sql = "SELECT MIN(nilai) AS
MIN FROM hasil";
        $stmt = $this->db-
>prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
        return $data;
    }

    public function hitung($val = 0){
        $no1 = 1;
        $sumc1 = 0;
        $sumc2 = 0;
        $sumc3 = 0;
        $sumc4 = 0;
        $sumc5 = 0;
        $forutilty = array(array());
        $foralt = array(array());
    }

```

```

$arraykey = 0;

$hasil = new Hasil();
$a = $hasil->count();
foreach($a as $row) {
    $b = $row['count'];
}
do{
    $hasil = new Hasil();
    $roww=$hasil->spk($sno1);
    if($roww== NULL){goto skip;}
    foreach($roww as $rows) {
        //Selection
        $jamban = $rows['c1'];
        $rumah = $rows['c2'];
        $penghuni = $rows['c3'];
        $sosial = $rows['c4'];
        $lahan = $rows['c5'];
        $alternatif = $rows['id'];
        $name = $rows['nama'];
        if($jamban=="Tidak Ada"){
            $nilai1 = 100;
        }else{
            $nilai1 = 50;
        }
        if($rumah=="Non Permanen"){

```

```

            $nilai2 = 100;
        }elseif($rumah=="Semi
Permanen"){
            $nilai2 = 75;
        }else{
            $nilai2 = 50;
        }
        if($penghuni>9){
            $nilai3 = 100;
        }elseif($penghuni>=7){
            $nilai3 = 90;
        }elseif($penghuni>=4){
            $nilai3 = 80;
        }else{
            $nilai3 = 70;
        }
        if($sosial=="Kurang Mampu"){
            $nilai4 = 100;
        }elseif($sosial=="Mampu"){
            $nilai4 = 75;
        }else{
            $nilai4 = 50;
        }
        if($lahan=="Tidak Ada"){
            $nilai5 = 50;

```



```

    }else{
        $nilai5 = 100;
    }
}

if($nilai1==NULL){goto skip;}

$C1= 100*(($nilai1 - 50)/(100-50));

$C2= 100*(($nilai2 - 50)/(100-50));

$C3= 100*(($nilai3 - 70)/(100-70));

$C4= 100*(($nilai4 - 50)/(100-50));

$C5= 100*(($nilai5 - 50)/(100-50));

//Tester Variabel

//echo ""Penghuni : ".$penghuni."
Nilai : ". $nilai3 ." Utility : ". $C3 .
'<br>';

$sumc1 = $sumc1 + $C1;

$sumc2 = $sumc2 + $C2;

$sumc3 = $sumc3 + $C3;

$sumc4 = $sumc4 + $C4;

$sumc5 = $sumc5 + $C5;

$forallt[$arraykey][0] =
Saltname;

for($i=1;$i<=5;$i++)
{

```

```

        $forallt[$arraykey][$i] =
        ${'nilai'.$i};
    }

    $forutilty[$arraykey][0] =
    Saltname;

    for($i=1;$i<=5;$i++)
    {
        $forutilty[$arraykey][$i] =
        ${'C'.$i};
    }
    $arraykey++;

    skip:
    $nilai1 =null;
    $no1++;
} while($no1 <= $b);

$sqrtc1 = sqrt($sumc1);
$sqrtc2 = sqrt($sumc2);
$sqrtc3 = sqrt($sumc3);
$sqrtc4 = sqrt($sumc4);
$sqrtc5 = sqrt($sumc5);

$data = array(
    '1' => $sqrtc1,
    '2' => $sqrtc2,
    '3' => $sqrtc3,
    '4' => $sqrtc4,

```

```

        '5' => $sqrtc5,
    );
    if($val== 0)
    {
        return $data;
    }elseif($val == 1){
        return $foralt;
    }else{
        return $forutilty;
    }
}
public function hitung2()
{
    $hasil = new Hasil();
    $akaran = $hasil->hitung();
    for($i=1;$i <=5; $i++)
    {
        ${'akarc'.$i} = $akaran[$i];
    }
    $no1= 1;
    $data = array(array());
    $hasil = new Hasil();
    $a = $hasil->count();
    foreach($a as $row) {
        $b = $row['count'];
    }
    $arraykey = 0;
    do{
        $hasil = new Hasil();
        $roww=$hasil->spk($no1);
        if($roww== NULL){goto skip;}
        foreach($roww as $rows) {
            //Selection
            $jamban = $rows['c1'];
            $rumah = $rows['c2'];
            $penghuni = $rows['c3'];
            $sosial = $rows['c4'];
            $lahan = $rows['c5'];
            $alternatif = $rows['id'];
            $altnama = $rows['nama'];
            if($jamban=="Tidak Ada"){
                $nilai1 = 100;
            }else{
                $nilai1 = 50;
            }
            if($rumah=="Non Permanen"){
                $nilai2 = 100;
            }elseif($rumah=="Semi
Permanen"){
                $nilai2 = 75;
            }
        }
    }while($arraykey < $a);
}

```

```

}else{
    $nilai2 = 50;
}
if($penghuni>9){
    $nilai3 = 100;
}elseif($penghuni>=7){
    $nilai3 = 90;
}elseif($penghuni>=4){
    $nilai3 = 80;
}
}else{
    $nilai3 = 70;
}
if($sosial=="Kurang Mampu"){
    $nilai4 = 100;
}elseif($sosial=="Mampu"){
    $nilai4 = 75;
}
}else{
    $nilai4 = 50;
}
if($lahan=="Tidak Ada"){
    $nilai5 = 50;
}
}
}

if($nilai1==NULL){goto skip;}
$C1= 100*(($nilai1 - 50)/(100-50));
$C2= 100*(($nilai2 - 50)/(100-50));
$C3= 100*(($nilai3 - 70)/(100-70));
$C4= 100*(($nilai4 - 50)/(100-50));
$C5= 100*(($nilai5 - 50)/(100-50));
for($i=1;$i <=5; $i++)
{
    ${'normalc'.$i} = ${'C'.$i} / ${'akarc'.$i};
}
for($i=1;$i <=6; $i++)
{
    if($i==6){
        $data[$arraykey][$i] = $alternatif;
        $data[$arraykey][$i+1] = $altnama;
    }else{
        $data[$arraykey][$i] = ${'normalc'.$i};
    }
}
}
}

```

```

    $arraykey++;
    //Tester Variabel
    //echo "Penghuni : '$penghuni.'
    Nilai : '" . $nilai3 ." Utility : '" . $C3 .
    '<br>";
    skip:
    $nilai1 =null;
    $no1++;
    } while($no1 <= $b);
    //Normalisasi
    return $data;
}
public function hitung3($set = 0)
{
    $hasil = new Hasil();
    $p1 = 25;
    $p2 = 15;
    $p3 = 15;
    $p4 = 20;
    $p5 = 25;
    for($i=1;$i <=5; $i++)
    {
        ${'bobotc'. $i} = ${'p'. $i}/100;
    }
    for($i=1;$i <=5; $i++)
    {

```

```

        ${'sumnormal'. $i} = 0.01;
    }
    $data = array(array());
    $akarnormal = array();
    $arraykey = 0;
    $normal = $hasil->hitung2();
    foreach($normal as $a)
    {
        for($i=1;$i <=6; $i++)
        {
            if($i==6){
                $data[$arraykey][$i] =
                $a[$i];
                $data[$arraykey][$i+1]
                = $a[$i+1];
            }else{
                $data[$arraykey][$i] =
                $a[$i]*${'bobotc'. $i};
                ${'normalbobot'. $i} =
                $a[$i]*${'bobotc'. $i};
                if(${'normalbobot'. $i}>${'sumnorma
                l'. $i})
                {
                    ${'sumnormal'. $i} =
                    ${'normalbobot'. $i};
                }
            }
        }
    }
}

```

```

        $arraykey++;
    }
    for($i=1;$i <=5; $i++)
    {
        $akarnormal[] =
    ${'sumnormal'. $i};
    }
    if($set==0){
        return $data;
    }else{
        return $akarnormal;
    }
}

public function hitung4($set = 0)
{
    $hasil = new Hasil();
    $dpostif = $hasil->hitung3();
    $dakaran = $hasil->hitung3(1);
    for($i=0;$i<5; $i++)
    {
        ${'akarnormal'. $i} =
    $dakaran[$i];
    }
    $data = array(array());

    $arraykey = 0;
    foreach($dpostif as $a)
    {
        $scal = 0;
        $scalminus = 0;
        for($i=1;$i <=6; $i++)
        {
            if($i==6){
                $data[$arraykey][2] =
    $a[$i];
                $dataminus[$arraykey][1] = $a[$i];

                $data[$arraykey][7] =
    $a[$i+1];
            }else{
                $num = $a[$i]-
    ${'akarnormal'.($i-1)};

                $scal = $scal +
    pow($num,2);

                $scalminus =
    $scalminus + pow($a[$i],2);
            }
        }

        $data[$arraykey][0] =
    sqrt($scal);

        $data[$arraykey][1] =
    sqrt($scalminus);

        $arraykey++;
    }
}

```

```

    }
    return $data;
}
public function preferensi()
{
    $hasil = new Hasil();
    $dpostif = $hasil->hitung4();
    foreach($dpostif as $a)
    {
        $prep = $a[1]/($a[1]+$a[0]);
        $value = round($prep,3);
        //Execute
        $sql = "INSERT INTO hasil
VALUES (:alternatif, :nilai)";
        $stmt = $this->db-
>prepare($sql);
        $stmt-
>bindParam(":alternatif", $a[2]);
        $stmt->bindParam(":nilai",
$value);
        $stmt->execute();
    }
}
public function ranking()
{
    $sql = "SELECT * FROM `hasil`,
alternatif WHERE hasil.id_alternatif

```

```

= alternatif.id ORDER BY
hasil.nilai DESC;";
    $stmt = $this->db->prepare($sql);
    $stmt->execute();
    $data = [];
    while ($rows = $stmt->fetch()) {
        $data[] = $rows;
    }
    return $data;
}
public function hapus()
{
    $sql = "DELETE FROM hasil ";
    $stmt = $this->db->prepare($sql);
    $stmt->execute();
}
public function findingnemo($set
= 0)
{
    $hasil = new Hasil();
    $dpostif = $hasil->hitung3();
    $dakaran = $hasil->hitung3(1);
    for($i=0;$i<5; $i++)
    {
        ${'akarnormal'.$i} =
$dakaran[$i];
    }
}

```

```

    }
    $data = array(array());
    $arraykey = 0;
    foreach($dpostif as $a)
    {
        if($a[6] == 4)
        {
            $scal = 0;
            $scalminus = 0;
            for($i=1;$i<6;$i++)
            {
                $num = $('akarnormal' . ($i-
1))- $a[$i];
                $scal = $scal +
pow($num,2);
                $scalminus = $scalminus +
pow($a[$i],2);
            }
            $data[$arraykey][0] =
$scal;
            $arraykey++;
            //$data[$arraykey][1] =
round(sqrt($scalminus),3);
        }
    }
    return $data;
}

```

```

}

```

```

?>

```

3. Kriteria.php

```

<?php
namespace App;
use PDO;
include "Koneksi.php";
class Kriteria extends Koneksi{
    public function __construct()
    {
        $call = new Koneksi();
        $this->db = $call->koneksi();
    }
    public function count()
    {
        $sql = "SELECT
COUNT(id_kriteria) AS count
FROM kriteria";
        $stmt = $this->db-
>prepare($sql);
        $stmt->execute();
        $data = [];
        while ($rows = $stmt->fetch())
        {
            $data[] = $rows;
        }
    }
}

```

```

return $data;
}
public function tampil()
{
    $sql = "SELECT * FROM kriteria
ORDER BY kriteria ASC";

    $stmt = $this->db->prepare($sql);
    $stmt->execute();

    $data = [];
    while ($rows = $stmt->fetch()) {
        $data[] = $rows;
    }
    return $data;
}

public function hapus()
{
    $sql = "SELECT * FROM kriteria
WHERE id_kriteria = :id";

    $stmt = $this->db->prepare($sql);
    $stmt->bindParam(":id",
$_GET['id']);

    $stmt->execute();

    $sql = "DELETE FROM
kriteria WHERE id_kriteria = :id";

    $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":id",
$_GET['id']);

        $stmt->execute();

        header("location: kriteria.php");
    }

    public function update()
    {
        $sql = "SELECT * FROM
kriteria WHERE id_kriteria = :id";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":id",
$_GET['id']);

        $stmt->execute();

        $sql = "UPDATE kriteria
SET kriteria = :kode, bobot = :bobot
, keterangan = :keterangan WHERE
id_kriteria = :id";

        $stmt = $this->db-
>prepare($sql);

        $stmt -> bindParam(":id",
$_GET['id']);

        $stmt ->
bindParam(":kode",
$_POST['kriteria']);

        $stmt ->
bindParam(":bobot",
$_POST['bobot']);

        $stmt ->
bindParam(":keterangan",
$_POST['keterangan']);

```



```

        $stmt->execute();
    }

    public function tambah()
    {
        $sql = "INSERT INTO kriteria
VALUES (:, :kode, :keterangan,
: bobot, :jenis)";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":kode",
$_POST['kode']);

        $stmt-
>bindParam(":keterangan",
$_POST['keterangan']);

        $stmt->bindParam(":bobot",
$_POST['bobot']);

        $stmt->bindParam(":jenis",
$_POST['jenis']);

        $stmt->execute();

        header("location: kriteria.php");
    }

    public function simpan()
    {
        $sql = "INSERT INTO kriteria
VALUES (:, :kriteria,
:keterangan, :bobot)";

        $stmt = $this->db-
>prepare($sql);

```

```

        $stmt->bindParam(":kriteria",
$_POST['kriteria']);

        $stmt-
>bindParam(":keterangan",
$_POST['keterangan']);

        $stmt->bindParam(":bobot",
$_POST['bobot']);

        $stmt->execute();

        header("location: kriteria.php");
    }

    public function get_json($val)
    {
        $sql = "SELECT * FROM
kriteria WHERE id_kriteria=:id limit
1";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":id", $val);

        $stmt->execute();

        //$data['data'] = array();

        $data = $stmt-
>fetchAll(PDO::FETCH_ASSOC);

        return json_encode($data);
    }
}
?>

```

4. Login.php

```
<?php
namespace App;
use PDO;
include "Koneksi.php";
class Login extends Koneksi{
    public function __construct()
    {
        $call = new Koneksi();
        $this->db = $call->koneksi();
    }
    public function login()
    {
        $sql = "SELECT * From
tbl_login WHERE username =
:user_nama AND password =
md5(:user_pass)";

        //query mencari datanya

        $stmt = $this->db-
>prepare($sql);

        $stmt-
>bindParam(":user_nama",
$_POST['user']); //statement bisa
diganti apa aja,

        $stmt-
>bindParam(":user_pass",
$_POST['pass']);

        $stmt->execute();

        $login = $stmt->fetch();
        if ($stmt->rowCount() == 0) {
            echo '<script>alert("Login
Gagal, Periksa Kembali Username
dan Password");
            window.location.href =
"index.php"</script>'; //tidak ada
datanya
        } else {
            session_start();
            $_SESSION["ID"] =
$login['id_login'];
            $_SESSION["nama"] =
$login['username'];
            $_SESSION["namapengguna"] =
$login['nama'];
            $_SESSION["status"] =
$login['level'];

            //FOR LOGs
            $str1 = " Melakukan Login
";

            $sql = "INSERT INTO
logs (id_log, keterangan, id_login)
VALUES (:, :user :txt, :id_login)";

            $stmt = $this->db-
>prepare($sql);

            $stmt->bindParam(":txt",
$str1);

            $stmt-
>bindParam(":id_login",
$login['id_login']);
```

```

        $stmt->bindParam(":user",
$login['username']);

        $stmt->execute();

        header("location:
index.php");

        exit;
    }
}

public function logout()
{
    $sql = "SELECT * From
tbl_login WHERE id_login = :id";
    //query mencari datanya

    $stmt = $this->db-
>prepare($sql);

    $stmt->bindParam(":id",
$_GET['id']); //statement bisa diganti
apa aja

    $stmt->execute();

    $login = $stmt->fetch();

    if ($stmt->rowCount() == 0) {

        echo '<script>alert("Login
Gagal, Periksa Kembali Username
dan Password");

        window.location-
"index.php"</script>'; //tidak ada
datanya

    } else {

        //FOR LOGs

        $str1 = " Telah Logout ";

        $sql = "INSERT INTO
logs (id_log, keterangan, id_login)
VALUES (:,user :txt, :id_login)";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":txt",
$str1);

        $stmt-
>bindParam(":id_login",
$login['id_login']);

        $stmt->bindParam(":user",
$login['username']);

        $stmt->execute();

        session_destroy();

        //header("location:
login.php");

        //exit;
    }
}

public function register()
{
    $sql = "SELECT * From
tbl_login WHERE username = :id";

    //query mencari datanya

    $stmt = $this->db-
>prepare($sql);

```

```

        $stmt->bindParam(":id",
$_POST['user1']); //statement bisa
diganti apa aja

        $stmt->execute();

        if ($stmt->rowCount() > 0) {

            echo
'<script>alert("Username Sudah
Digunakan");

            window.location-
"login.php"</script>'; //tidak ada
datanya

        } else {

            $sql = "INSERT INTO
tbl_login VALUES (:user_nama,
:nama, md5(:user_pass), 'User')";

            $stmt = $this->db-
>prepare($sql);

            $stmt-
>bindParam(":user_nama",
$_POST['user1']);

            $stmt->bindParam(":nama",
$_POST['nama']);

            $stmt-
>bindParam(":user_pass",
$_POST['pass1']);

            $stmt->execute();

            echo '<script>alert("Success,
User Berhasil Dibuat");

            window.location-
"login.php"</script>';

        }

```

```

    }
}
?>

```

5. Penilaian.php

```

<?php
namespace App;

use PDO;

include "Koneksi.php";

class Penilaian extends Koneksi{

    public function __construct()
    {
        $call = new Koneksi();

        $this->db = $call->koneksi();

    }

    public function update()
    {

        $sql = "SELECT * FROM
alternatif WHERE id = :id";

        $stmt = $this->db-
>prepare($sql);

        $stmt->bindParam(":id",
$_GET['id']);

        $stmt->execute();

        $array = $stmt->fetch();

```

```

        $sql = "UPDATE alternatif
SET c1 = :c1, c2 = :c2, c3 = :c3, c4 =
:c4, c5 = :c5 WHERE id = :id";

```

```

        $stmt = $this->db-
>prepare($sql);

```

```

        $stmt -> bindParam(":id",
$_GET['id']);

```

```

        $stmt -> bindParam(":c1",
$_POST['c1']);

```

```

        $stmt -> bindParam(":c2",
$_POST['c2']);

```

```

        $stmt -> bindParam(":c3",
$_POST['c3']);

```

```

        $stmt -> bindParam(":c4",
$_POST['c4']);

```

```

        $stmt -> bindParam(":c5",
$_POST['c5']);

```

```

        $stmt->execute();

```

```

    }

```

```

    public function tampil()

```

```

    {

```

```

        $sql = "SELECT * FROM
alternatif ";

```

```

        $stmt = $this->db->prepare($sql);

```

```

        $stmt->execute();

```

```

        $data = [];

```

```

        while ($rows = $stmt->fetch()) {

```

```

            $data[] = $rows;

```

```

        }

```

```

        return $data;

```

```

    }

```

```

    public function get_json($val)

```

```

    {

```

```

        $sql = "SELECT * FROM
alternatif WHERE id=:id limit 1";

```

```

        $stmt = $this->db-
>prepare($sql);

```

```

        $stmt->bindParam(":id", $val);

```

```

        $stmt->execute();

```

```

        //$data['data'] = array();

```

```

        $data = $stmt-
>fetchAll(PDO::FETCH_ASSOC);

```

```

        return json_encode($data);

```

```

    }

```

```

}

```

```

?>

```

6. Perhitungan.php

```

<?php

```

```

namespace App;

```

```

use PDO;

```

```

require_once "Users.php";

```

```

class Perhitungan{

```

```

    private $db;

```

```

    public function __construct()

```

```

{
try {
$this->db =
new
PDO("mysql:host=localhost;dbname
=dinkes","root","");
} catch (PDOException $e) {
die ("Error " . $e->getMessage());
}
}
public function tampil()
{
    $sql = "SELECT * FROM
alternatif";
    $stmt = $this->db->prepare($sql);
    $stmt->execute();
    $data = [];
    while ($rows = $stmt->fetch()) {
        $data[] = $rows;
    }
    return $data;
}
public function nakes($a)
{
    if($a>=85){
        $nilai = 5;
    }elseif($a>=64){
        $nilai = 4;
    }elseif($a>= 43){
        $nilai = 3;
    }elseif($a>= 22){
        $nilai = 2;
    }else{
        $nilai = 1;
    }
}
public function mpv($a)
{
    if($a>=8){
        $nilai = 5;
    }elseif($a>=6){
        $nilai = 4;
    }elseif($a>=4){
        $nilai = 3;
    }elseif($a>=2){
        $nilai = 2;
    }else{
        $nilai = 1;
    }
}
}

```

```

        return $nilai;
    }

    public function izin_ops($a)
    {
        if($a=='Sudah Ada'){
            $nilai = 5;
        }else{
            $nilai = 1;
        }
        return $nilai;
    }

    public function luas_ruang($a)
    {
        if($a>40){
            $nilai = 5;
        }elseif($a>30){
            $nilai = 4;
        }elseif($a>20){
            $nilai = 3;
        }elseif($a>10){
            $nilai = 2;
        }else{
            $nilai = 1;
        }
    }

```

```

        return $nilai;
    }

    public function internet($a)
    {
        if($a>=25){
            $nilai = 5;
        }elseif($a>=19){
            $nilai = 4;
        }elseif($a>= 13){
            $nilai = 3;
        }elseif($a>= 7){
            $nilai = 2;
        }else{
            $nilai = 1;
        }
        return $nilai;
    }
}
?>

```

7. Subkriteria.php

```

<?php
namespace App;
use PDO;
class Subkriteria extends Koneksi{

```

```

public function __construct()
{
    $call = new Koneksi();
    $this->db = $call->koneksi();
}
public function count()
{
    $sql = "SELECT
COUNT(id_sub) AS count
FROM sub_kriteria";
    $stmt = $this->db-
>prepare($sql);
    $stmt->execute();
    $data = [];
    while ($rows = $stmt->fetch())
    {
        $data[] = $rows;
    }
    return $data;
}
public function kriteria()
{
    $sql = "SELECT * FROM
kriteria";
    $stmt = $this->db-
>prepare($sql);
    $stmt->execute();
    $data = [];
    while ($rows = $stmt->fetch())
    {
        $data[] = $rows;
    }
}
}
return $data;
}
public function tampil($a)
{
    $sql = "SELECT * FROM
sub_kriteria WHERE id_kriteria
= :id ORDER BY nilai ASC";
    $stmt = $this->db->prepare($sql);
    $stmt->bindParam(":id", $a);
    $stmt->execute();
    $data = [];
    while ($rows = $stmt->fetch()) {
        $data[] = $rows;
    }
    return $data;
}
public function tampilkrit()
{
    $sql = "SELECT * FROM
kriteria";
    $stmt = $this->db->prepare($sql);
    $stmt->execute();
    $data = [];
    while ($rows = $stmt->fetch()) {
        $data[] = $rows;
    }
    return $data;
}
}
public function hapus()

```



```

{
    $sql = "SELECT * FROM
        sub_kriteria WHERE id_sub =
        :id";
    $stmt = $this->db->prepare($sql);
    $stmt->bindParam(":id",
        $_GET['id']);
    $stmt->execute();
    $sql = "DELETE FROM
        sub_kriteria WHERE id_sub =
        :id";
    $stmt = $this->db-
        >prepare($sql);
    $stmt->bindParam(":id",
        $_GET['id']);
    $stmt->execute();
    header("location:
        subkriteria.php");
}

public function update()
{
    $sql = "SELECT * FROM
        sub_kriteria WHERE id_sub =
        :id";
    $stmt = $this->db-
        >prepare($sql);
    $stmt->bindParam(":id",
        $_GET['id']);
    $stmt->execute();
    $sql = "UPDATE sub_kriteria
        SET keterangan = :keterangan
        ,nilai = :nilai WHERE id_sub =
        :id";
    $stmt = $this->db-
        >prepare($sql);
    $stmt -> bindParam(":id",
        $_GET['id']);
    $stmt -> bindParam(":keterangan",
        $_POST['keterangan']);
    $stmt -> bindParam(":nilai",
        $_POST['nilai']);
    $stmt->execute();
}

public function tambah()
{
    $sql = "INSERT INTO
        sub_kriteria VALUES (:kode,
        :keterangan, :nilai)";
    $stmt = $this->db-
        >prepare($sql);
    $stmt->bindParam(":kode",
        $_POST['kriteria']);
    $stmt-
        >bindParam(":keterangan",
        $_POST['keterangan']);
    $stmt->bindParam(":nilai",
        $_POST['nilai']);
    $stmt->execute();
    header("location:
        subkriteria.php");
}

```

```

public function simpan()
{
    $sql = "INSERT INTO
sub_kriteria VALUES (',
:keterangan, :nilai,
:id_kriteria)";
$stmt = $this->db-
>prepare($sql);
$stmt-
>bindParam(":id_kriteria",
$_POST['kriteria']);
$stmt-
>bindParam(":keterangan",
$_POST['keterangan']);
$stmt->bindParam(":nilai",
$_POST['nilai']);
$stmt->execute();
header("location:
subkriteria.php");
}

public function get_json($val)
{
    $sql = "SELECT * FROM
sub_kriteria WHERE
id_sub=:id limit 1";
$stmt = $this->db-
>prepare($sql);
$stmt->bindParam(":id", $val);
$stmt->execute();
}
}

// $data['data'] = array();
$data = $stmt-
>fetchAll(PDO::FETCH ASSO
C);
return json_encode($data);
}
?>

```