

DAFTAR LAMPIRAN

LAMPIRAN 1:

Alternatif Pemilihan

Alternatif	Kriteria				
	C_1	C_2	C_3	C_4	C_5
A1	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A2	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A3	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A4	Tidak	Ya	Ya	≤Rp1.000.000,-	Non Permanent
A5	Ya	Ya	Ya	≤Rp1.000.000,-	Permanent
A6	Ya	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A7	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A8	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A9	Tidak	Ya	Ya	≤Rp1.000.000,-	Non Permanent
A10	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A11	Ya	Ya	Ya	≤Rp1.000.000,-	Non Permanent
A12	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A13	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A14	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A15	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A16	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A17	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent

A18	Tidak	Ya	Ya	≤Rp1.000.000,-	Non Permanent
A19	Ya	Ya	Ya	≤Rp1.000.000,-	Non Permanent
A20	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A21	Ya	Ya	Ya	≤Rp1.000.000,-	Non Permanent
A22	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A23	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A24	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A25	Ya	Tidak	Tidak	>Rp1.000.000,-	Non Permanent
A26	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A27	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A28	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A29	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A30	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A31	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A32	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A33	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A34	Ya	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A35	Tidak	Tidak	Ya	≤Rp1.000.000,-	Non Permanent
A36	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent

A37	Ya	Ya	Ya	>Rp1.000.000,-	Permanent
A38	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A39	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A40	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A41	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A42	Ya	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A43	Ya	Tidak	Ya	>Rp1.000.000,-	Permanent
A44	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A45	Tidak	Tidak	Ya	≤Rp1.000.000,-	Non Permanent
A46	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A47	Tidak	Tidak	Ya	≤Rp1.000.000,-	Permanent
A48	Tidak	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A49	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A50	Tidak	Ya	Tidak	≤Rp1.000.000,-	Permanent
A51	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A52	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A53	Ya	Tidak	Ya	≤Rp1.000.000,-	Non Permanent
A54	Ya	Tidak	Ya	≤Rp1.000.000,-	Permanent
A55	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A56	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A57	Tidak	Ya	Ya	>Rp1.000.000,-	Permanent
A58	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent

A59	Ya	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A60	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A61	Tidak	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A62	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A63	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A64	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A65	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A66	Tidak	Ya	Tidak	≤Rp1.000.000,-	Non Permanent
A67	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A68	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A69	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A70	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A71	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A72	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A73	Tidak	Tidak	Tidak	>Rp1.000.000,-	Non Permanent
A74	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A75	Ya	Tidak	Tidak	>Rp1.000.000,-	Non Permanent
A76	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A77	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A78	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A79	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A80	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A81	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent

A82	Tidak	Tidak	Tidak	>Rp1.000.000,-	Non Permanent
A83	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A84	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A85	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A86	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A87	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A88	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A89	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A90	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A91	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A92	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A93	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A94	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A95	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A96	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Permanent
A97	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A98	Ya	Ya	Tidak	>Rp1.000.000,-	Permanent
A99	Ya	Tidak	Tidak	>Rp1.000.000,-	Permanent
A100	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A101	Tidak	Tidak	Tidak	>Rp1.000.000,-	Non Permanent

A102	Tidak	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A103	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A104	Tidak	Ya	Tidak	>Rp1.000.000,-	Permanent
A105	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A106	Tidak	Ya	Tidak	>Rp1.000.000,-	Permanent
A107	Tidak	Ya	Tidak	>Rp1.000.000,-	Permanent
A108	Ya	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A109	Ya	Ya	Tidak	≤Rp1.000.000,-	Non Permanent
A110	Tidak	Tidak	Tidak	>Rp1.000.000,-	Non Permanent
A111	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A112	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A113	Tidak	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A114	Tidak	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A115	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent
A116	Tidak	Ya	Tidak	>Rp1.000.000,-	Permanent
A117	Tidak	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent
A118	Ya	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A119	Ya	Tidak	Tidak	≤Rp1.000.000,-	Non Permanent

A120	Tidak	Ya	Tidak	>Rp1.000.000,-	Non Permanent
A121	Tidak	Tidak	Tidak	>Rp1.000.000,-	Permanent



THE
Character Building
UNIVERSITY

LAMPIRAN 2:

Source Code WEB

```
//WASPASController.php
<?php

namespace App\Http\Controllers;

use App\Models\DataBantuan;
use Illuminate\Http\Request;

class WASPASController extends Controller
{
    const weight = [
        0.4567, // Memiliki kartu jaminan gratis dari pemerintah
        0.2567, // Mengalami kehilangan mata pencaharian
        0.1567, // Mempunyai anggota keluarga yang rentang sakit
        0.09, // Pendapatan kepala rumah tangga
        0.04, // Jenis rumah
    ];

    const criteriaList = [
        "benefit", // Memiliki kartu jaminan gratis dari pemerintah
        "benefit", // Mengalami kehilangan mata pencaharian
        "benefit", // Mempunyai anggota keluarga yang rentang sakit
        "benefit", // Pendapatan kepala rumah tangga
        "benefit", // Jenis rumah
    ];

    function transpose($matrix)
    {
        $array = [];
        foreach ($matrix as $row => $columns) {
            foreach ($columns as $row2 => $column2) {
                $array[$row2][$row] = $column2;
            }
        }
        return $array;
    }

    function findMaxFromArray($array)
    {
        $max = $array[0];
    }
}
```

```

        foreach ($array as $value) {
            if ($value > $max) {
                $max = $value;
            }
        }
        return $max;
    }

    function findMinFromArray($array)
    {
        $min = $array[0];
        foreach ($array as $value) {
            if ($value < $min) {
                $min = $value;
            }
        }
        return $min;
    }

    function normalizeByCriteria($array, $criteria)
    {
        if ($criteria === "benefit") {
            $max = $this->findMaxFromArray($array);
            return array_map(function ($value) use ($max) {
                return ($value / $max);
            }, $array);
        }

        $min = $this->findMinFromArray($array);
        return array_map(function ($value) use ($min) {
            return ($min / $value);
        }, $array);
    }

    function countQiValue($array)
    {
        $sigma = 0.0;
        $product = 1.0;

        foreach ($array as $index => $value) {
            $sigma += $value * self::weight[$index];
            $product *= pow($value, self::weight[$index]);
        }

        return ($sigma * 0.5 + $product * 0.5);
    }

```

```

}

public function getBantuan(Request $request)
{
    $bantuan = DataBantuan::all();
    $names = [];
    $data = [];

    foreach ($bantuan as $b) {
        $names[] = $b['nama_kepala_keluarga'];
        $data[] = [
            $b['memiliki_kartu_gratis'],
            $b['kehilangan_pencaharian'],
            $b['anggota_keluarga_kronis'],
            $b['pendapatan_keluarga'],
            $b['jenis_rumah'],
        ];
    }

    $dataT = $this->transpose($data);

    $normalizedData = [];
    for ($i = 0; $i < count($dataT); $i++) {
        $criteria = self::criterialist[$i];
        $normalizedData[] = $this->normalizeByCriteria($dataT[$i],
$criteria);
    }

    $normalizedDataT = $this->transpose($normalizedData);
    $qiValues = [];
    foreach ($normalizedDataT as $row) {
        $qiValues[] = $this->countQiValue($row);
    }

    $currentName = $request->session()-
>get('credential')['nama_kepala_keluarga'];
    $sortedData = [];
    for ($i = 0; $i < count($qiValues); $i++) {
        $name = $names[$i];
        $qi = $qiValues[$i];

        $sortedData[] = [
            'name' => $name,
            'qi' => $qi,
        ];
    }
}

```

```

    }

    // sort sortedData by qi
    for ($i = 0; $i < count($sortedData); $i++) {
        for ($j = $i + 1; $j < count($sortedData); $j++) {
            if ($sortedData[$i]['qi'] < $sortedData[$j]['qi']) {
                $temp = $sortedData[$i];
                $sortedData[$i] = $sortedData[$j];
                $sortedData[$j] = $temp;
            }
        }
    }

    // iterate through sortedData, and check if index <= 63 flag
    layak to true else false
    for ($i = 0; $i < count($sortedData); $i++) {
        $name = $sortedData[$i]['name'];
        $qi = $sortedData[$i]['qi'];
        $layak = false;
        if ($i <= 63) {
            $layak = true;
        }

        DataBantuan::where('nama_kepala_keluarga', $name)
            ->update([
                'skor' => $qi,
                'layak' => $layak,
            ]);

        if ($name === $currentName) {
            $request->session()->put('credential',
            DataBantuan::where('nama_kepala_keluarga', $name)->first());
        }
    }

    return view('bantuan');
}
}

```

```

//AuthenticationController.php
<?php

```

```

namespace App\Http\Controllers;

```

```

use App\Models\DataBantuan;
use App\Models\DataPengguna;
use Illuminate\Http\Request;

class AuthenticationController extends Controller
{
    function signIn(Request $request)
    {
        $request->validate([
            'full_name' => 'required',
            'password' => 'required',
        ]);

        $user = DataPengguna::where('nama_lengkap',
$request['full_name'])
->first();

        if (!$user) {
            $request->session()->flash('error', [
                'code' => 401,
                'status' => 'unauthorized',
                'message' => 'Akun tidak ditemukan',
            ]);
            return redirect()->route('signin');
        }

        if (!password_verify($request->password, $user->password)) {
            $request->session()->flash('error', [
                'code' => 401,
                'status' => 'unauthorized',
                'message' => 'Password salah',
            ]);
            return redirect()->route('signin');
        }

        $data = DataBantuan::where('nama_kepala_keluarga',
$user['nama_lengkap'])
->first();

        $request->session()->put('credential', $data);
        return redirect()->route('profile');
    }

    function signUp(Request $request)

```

```

{
    $request->validate([
        'full_name' => 'required',
        'password' => 'required'
    ]);

    $isExist = DataPengguna::where('nama_lengkap',
$request['full_name'])
        ->count();
    if ($isExist) {
        $request->session()->flash('error', [
            'code' => 409,
            'status' => 'conflict',
            'message' => 'Akun dengan nama tersebut sudah
terdaftar',
        ]);
        return redirect()->route('signup');
    }

    DataPengguna::create([
        'nama_lengkap' => $request['full_name'],
        'password' => $request['password'],
    ]);

    $data = DataBantuan::create([
        'nama_kepala_keluarga' => $request['full_name'],
        'memiliki_kartu_gratis' => 0,
        'kehilangan_pencarian' => 0,
        'anggota_keluarga_kronis' => 0,
        'pendapatan_keluarga' => 0,
        'jenis_rumah' => 0,
    ]);

    $request->session()->put('credential', $data);
    return redirect()->route('profile');
}

function signOut(Request $request)
{
    $request->session()->forget('credential');
    return redirect()->route('onboarding');
}

function updateInfo(Request $request)
{

```

```

        $data = DataBantuan::where('nama_kepala_keluarga', $request->full_name)
            ->first();

        // check if data is null, if yes return error
        if (!$data) {
            return response()->json([
                'code' => 404,
                'status' => 'not_found',
                'message' => 'Data dengan nama tersebut tidak
ditemukan',
            ]);
        }

        // update data
        $data['memiliki_kartu_gratis'] =
$request['memiliki_kartu_gratis'];
        $data['kehilangan_pencaharian'] =
$request['kehilangan_pencaharian'];
        $data['anggota_keluarga_kronis'] =
$request['anggota_keluarga_kronis'];
        $data['pendapatan_keluarga'] = $request['pendapatan_keluarga'];
        $data['jenis_rumah'] = $request['jenis_rumah'];
        $data->save();

        $request->session()->put('credential', $data);
        return response()->json([
            'code' => 200,
            'status' => 'ok',
            'message' => 'Data berhasil diperbarui',
        ]);
    }
}

```

```

//DataBantuan.php
<?php

```

```

namespace App\Models;

```

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Notifications\Notifiable;

```

```

class DataBantuan extends Model
{
    use HasFactory, Notifiable;

    protected $table = 'data_bantuan';

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'nama_kepala_keluarga',
        'memiliki_kartu_gratis',
        'kehilangan_pencaharian',
        'anggota_keluarga_kronis',
        'pendapatan_keluarga',
        'jenis_rumah',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [];
}

```

```

//DataPengguna.php
<?php

```

```

namespace App\Models;

```

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasOne;
use Illuminate\Notifications\Notifiable;

class DataPengguna extends Model
{
    use HasFactory, Notifiable;

    protected $table = 'data_pengguna';

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'nama_lengkap',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'password' => 'hashed',
    ];
}

```

LAMPIRAN 3:

Surat Izin Penelitian



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS NEGERI MEDAN
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
Jl. Willem Iskandar Psr V - Medan Estate. Kotak Pos No. 1589 Medan 20221
www.fmipa.unimed.ac.id

Nomor : 3530 /UN33.4.1/PG/2023 Medan, 08 Juni 2023
 Lampiran : 1 (satu) berkas Proposal Penelitian
 Perihal : Izin Melaksanakan Penelitian

Yth. Kepala Desa Huta Limbong Kecamatan Padangsidempuan Tenggara
 Desa Huta Limbong
 di
 Tempat

Dengan hormat, kami memohon bantuan Saudara agar dapat memberikan izin melaksanakan Penelitian di instansi yang Saudara pimpin kepada mahasiswa tersebut di bawah ini :

Nama : Yolanda Feby
 NIM : 4193550008
 Program Studi : S-1 Ilmu Komputer
 Dosen Pembimbing : Dr. Armita, M.Si
 Judul Penelitian : Penerapan Metode WASPAS Dalam Penerima Bantuan Langsung Tunai-Dana Desa (BLT-Dana Desa) (Studi Kasus: Desa Huta Limbong Kecamatan Padangsidempuan Tenggara)
 Tempat Penelitian : Desa Huta Limbong Kecamatan Padangsidempuan Tenggara

Perlu diketahui bahwa kegiatan ini dilaksanakan untuk memperoleh data yang akan digunakan dalam penyusunan skripsi mahasiswa tersebut guna memenuhi salah satu syarat memperoleh gelar Sarjana Komputer (S.Kom) di FMIPA Unimed.

Demikian kami sampaikan, atas perhatian dan kerja sama yang baik diucapkan terima kasih.


 Dekan,
 Wakil Dekan Bidang Akademik
 Dr. Jamalun Purba, M.Si
 NIP. 19641207 199103 1 002



© Aplikasi SiManTep FMIPA Unimed - Dicitak Oleh : Sarah Agustina, S.Pd
 Pada hari, tanggal : Thursday, 08 June 2023 Jam : 15:54:44

LAMPIRAN 4:

Surat Balasan Izin Penelitian



**PEMERINTAH KOTA PADANGSIDIMPUAN
KECAMATAN PADANGSIDIMPUAN TENGGARA
DESA HUTA LIMBONG
Jl. Raja Imbang Desa**

Kode Pos : 22733

SURAT KETERANGAN

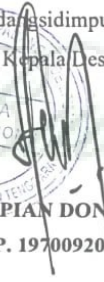
Nomor : 421.4 / 82 / 2011 / 2023

Sehubungan dengan surat dari Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Medan, Nomor: 3530 / UN33.4.1 / PG / 2023, hal: Izin Melaksanakan Penelitian tertanggal 08 Juni 2023, maka Kepala Desa Huta Limbong Kecamatan Padangsidempuan Tenggara dengan ini menerangkan nama mahasiswa di bawah ini :

Nama : Yolanda Feby
NIM : 4193550008
Program Studi : S-1 Ilmu Komputer
Dosen Pembimbing : Dr. Arnita, M.Si

Benar telah mengadakan penelitian di Desa Huta Limbong Kecamatan Padangsidempuan Tenggara guna melengkapi data pada penyusunan Skripsi yang berjudul : **“Penerapan Metode WASPAS Dalam Penerima Bantuan Langsung Tunai-Dana Desa (BLT-Dana Desa) (Studi Kasus: Desa Huta Limbong Kecamatan Padangsidempuan Tenggara)”**.

Demikian Surat Keterangan ini diperbuat untuk dapat dipergunakan seperlunya.

Padangsidempuan, 26 Juni 2023
Pj. Kepala Desa Huta Limbong,

SOPIAN DONGORAN, S.Sos
NIP. 197009202009061001