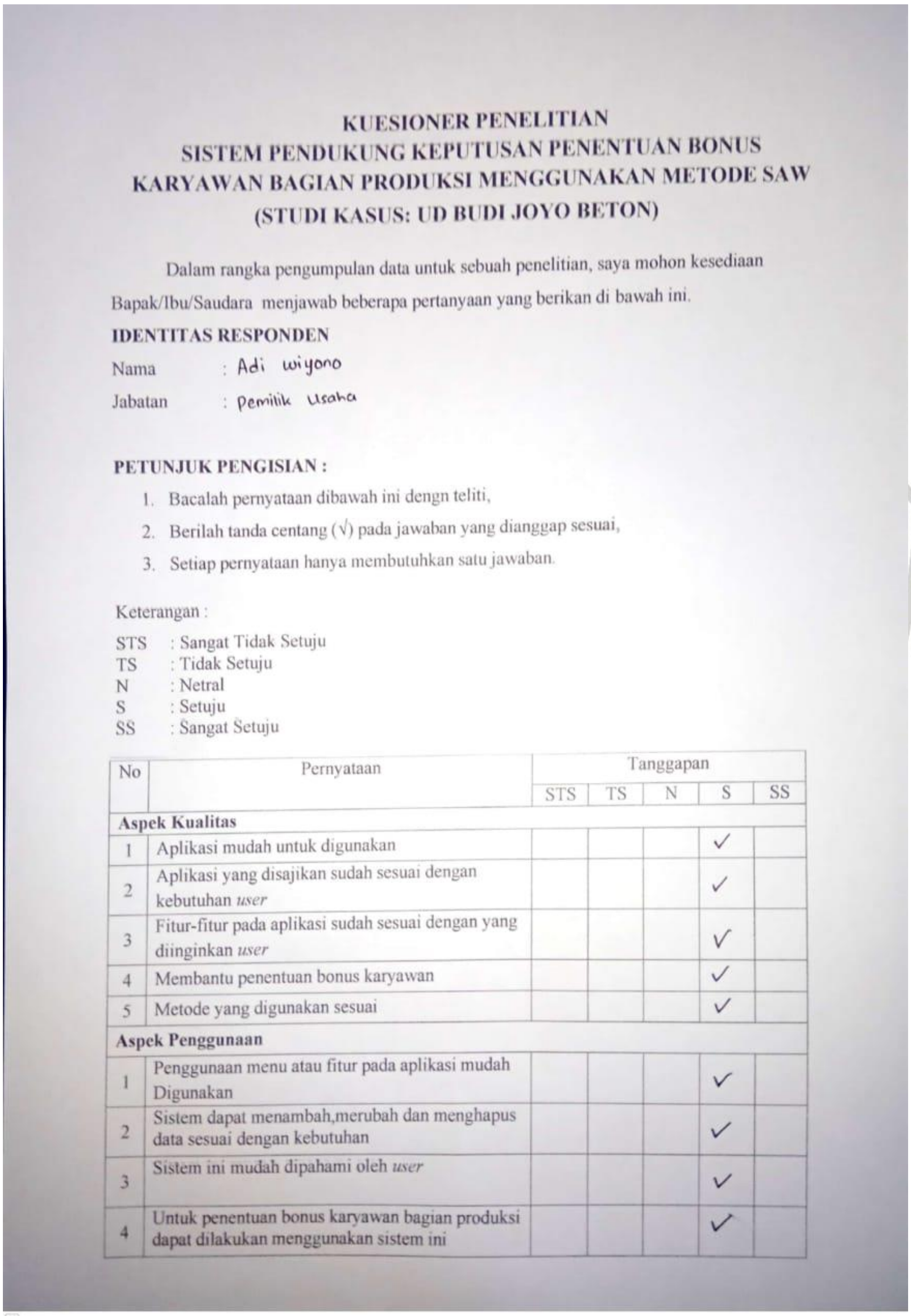
LAMPIRAN

Lampiran 1. Kuesioner Penelitian



KUESIONER PENELITIAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN BONUS KARYAWAN BAGIAN PRODUKSI MENGGUNAKAN METODE SAW (STUDI KASUS: UD BUDI JOYO BETON)

Dalam rangka pengumpulan data untuk sebuah penelitian, saya mohon kesediaan Bapak/Ibu/Saudara menjawab beberapa pertanyaan yang berikan di bawah ini.

IDENTITAS RESPONDEN

Nama : Adi wiyono

Jabatan : Pemilik Usaha

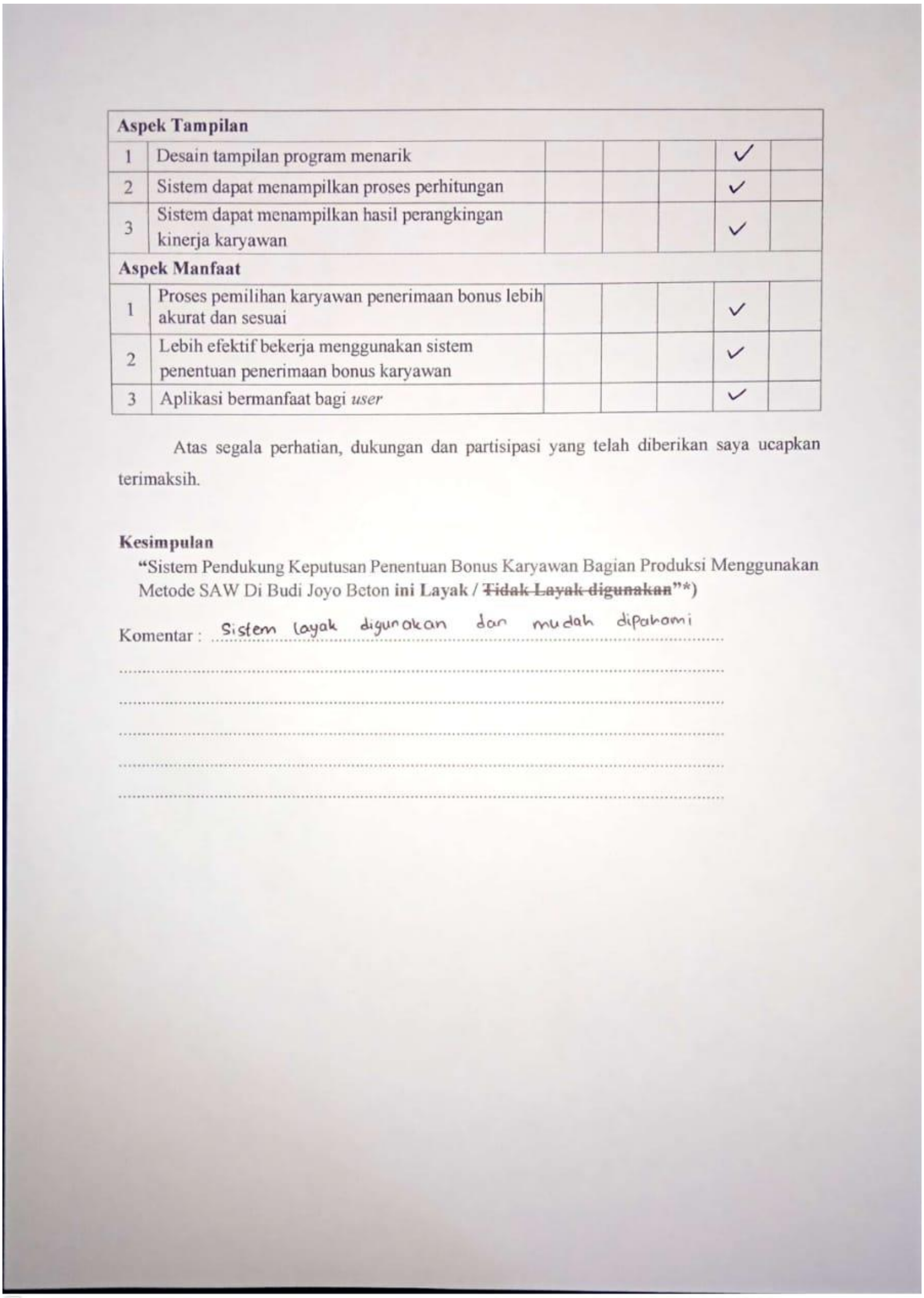
PETUNJUK PENGISIAN :

1. Bacalah pernyataan dibawah ini dengn teliti,
2. Berilah tanda centang (√) pada jawaban yang dianggap sesuai,
3. Setiap pernyataan hanya membutuhkan satu jawaban.

Keterangan :

STS	: Sangat Tidak Setuju
TS	: Tidak Setuju
N	: Netral
S	: Setuju
SS	: Sangat Setuju

No	Pernyataan	Tanggapan				
		STS	TS	N	S	SS
Aspek Kualitas						
1	Aplikasi mudah untuk digunakan				✓	
2	Aplikasi yang disajikan sudah sesuai dengan kebutuhan *user*				✓	
3	Fitur-fitur pada aplikasi sudah sesuai dengan yang diinginkan *user*				✓	
4	Membantu penentuan bonus karyawan				✓	
5	Metode yang digunakan sesuai				✓	
Aspek Penggunaan						
1	Penggunaan menu atau fitur pada aplikasi mudah Digunakan				✓	
2	Sistem dapat menambah,merubah dan menghapus data sesuai dengan kebutuhan				✓	
3	Sistem ini mudah dipahami oleh *user*				✓	
4	Untuk penentuan bonus karyawan bagian produksi dapat dilakukan menggunakan sistem ini				✓	



Aspek Tampilan						
1	Desain tampilan program menarik				✓	
2	Sistem dapat menampilkan proses perhitungan				✓	
3	Sistem dapat menampilkan hasil perangkingan kinerja karyawan				✓	
Aspek Manfaat						
1	Proses pemilihan karyawan penerimaan bonus lebih akurat dan sesuai				✓	
2	Lebih efektif bekerja menggunakan sistem penentuan penerimaan bonus karyawan				✓	
3	Aplikasi bermanfaat bagi *user*				✓	

Atas segala perhatian, dukungan dan partisipasi yang telah diberikan saya ucapkan terimaksih.

Kesimpulan

"Sistem Pendukung Keputusan Penentuan Bonus Karyawan Bagian Produksi Menggunakan Metode SAW Di Budi Joyo Beton **ini Layak / ~~Tidak Layak digunakan~~"*)**

Komentar : Sistem layak digunakan dan mudah dipahami

Lampiran 2. Source Code Sistem

1. AlternatifController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Alternatif;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class AlternatifController extends Controller
{
   // tampil data
   public function index()
   {
       $alternatifs = Alternatif::get();
       return view('pages.admin.alternatif.index', compact('alternatifs'));
   }
// create data
   public function store(Request $request)
   {
           $validate = $request->validate([
           'name' => 'nullable',
       ]);
 DB::beginTransaction();
       try {
          Alternatif::create($validate);
          DB::commit();
              return redirect()->back()->with('success', 'Data berhasil ditambahkan.');
       } catch (\Throwable $th) {
          DB::rollBack();
          return $th->getMessage();
       }
   }
// edit data
   public function update(Request $request, Alternatif $alternatif)
   {
       $validate = $request->validate([
          'name' => 'nullable'
       ]);
       DB::beginTransaction();
       try {
         $alternatif->update($validate);
          DB::commit();
              return redirect()->back()->with('success', 'Data berhasil diubah.');
           } catch (\Throwable $th) {
              DB::rollBack();
             return $th->getMessage();
           }
       }
// hapus data
    public function destroy(Alternatif $alternatif)
    {
        DB::beginTransaction();
        try {
            $alternatif->delete();
            DB::commit();
            return redirect()->back()->with('success', 'Data berhasil di hapus');
        } catch (\Throwable $th) {
            DB::rollBack();
            return $th->getMessage();
        }
    }
}
```

2. CripsController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Bobot;
use App\Models\Crips;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CripsController extends Controller
{
// tampilkan data
    public function index()
    {
        $crips = Crips::with('bobot')->orderBy('bobot_id')->get();
        $bobots = Bobot::get();
        return view('pages.admin.crips.index', compact('crips', 'bobots'));
    }
// tambah data
    public function store(Request $request)
    {
        DB::beginTransaction();
        try {
            Crips::create($request->all());
            DB::commit();
            return redirect()->back()->with('success', 'Data berhasil ditambahkan.');
        } catch (\Throwable $th) {
            DB::rollback();
            return $th->getMessage();
        }
    }


// edit data
    public function update(Request $request, Crips $crip)
    {
        DB::beginTransaction();
        try {
          $crip->update($request->all());
            DB::commit();
            return redirect()->back()->with('success', 'Data berhasil diupdate.');
        } catch (\Throwable $th) {
            DB::rollback();
            return $th->getMessage();
        }
    }
    // hapus data
    public function destroy(Crips $crip)
    {
        DB::beginTransaction();
        try {
            $crip->delete();
            DB::commit();
            return redirect()->back()->with('success', 'Data berhasil dihapus.');
        } catch (\Throwable $th) {
            DB::rollback();
            return $th->getMessage();
        }
```

3. EvaluasiController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Alternatif;
use App\Models\Bobot;
use App\Models\Crips;
use App\Models\Evaluasi;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class EvaluasiController extends Controller
{
// tampil data
    public function index()
    {
        $evaluasis = Evaluasi::with('alternatif', 'bobot', 'crips')->get();
        $alternatifs = Alternatif::get();
        $bobots = Bobot::with('crips')->get();
        // return $evaluasis;
        return view('pages.admin.evaluasi.index', compact('evaluasis', 'alternatifs', 'bobots'));
    }
 // tambah data

    public function store(Request $request)
    {
        // return $request;
        $validate = $request->validate([
            'alternatif_id' => 'nullable',
        ]);
        DB::beginTransaction();
        try {
            $date = Carbon::now()-
>month($request->month)-
>year($request->year)-
>translatedFormat('Y-m-d');
           $validate['tanggal'] = $da
te;
           $bobot = Bobot::count();
            for ($i = 0; $i < $bobot;
$i++) {
               if ($request['crips_' . $
i] !== null) {
                  $validate['bobot_id'
] = $request['bobot_' . $i];
                  $validate['crips_id']
= $request['crips_' . $i];
                  Evaluasi::create($v
alidate);
               }
           }
           DB::commit();
           return redirect()->back()-
>with('success', 'Data berhasil di
tambahkan.');
        } catch (\Throwable $th) {
           DB::rollback();
          return $th->getMessage();
        }
    }
```

```
    // edit edit data
     public function update(Reque
st $request, Evaluasi $evaluasi)
    {
        DB::beginTransaction();
        try {
                              $evaluasi-
>update($request->all());
            DB::commit();
            return redirect()->back()-
>with('success', 'Data berhasil di
update.');
        } catch (\Throwable $th) {
            DB::rollback();
           return $th->getMessage();
        }
    }
    // hapus data
     public function destroy(Evalu
asi $evaluasi)
    {
        DB::beginTransaction();
  try {
        $evaluasi->delete();
        DB::commit();
             return redirect()->back()-
>with('success', 'Data berhasil dihap
us.');
   } catch (\Throwable $th) {
        DB::rollback();
```

```
            return $th->getMessage();
        }
    }
    // function cari
    public function cari(Request $request)
    {
        $year = $request['year'] ?? date('Y');
        $month = $request['month'] ?? date('m');
        $evaluasis = Evaluasi::with('alternatif', 'bobot', 'crips')
        ->whereYear('tanggal',$year)
        ->whereMonth('tanggal',$month)
        ->get();
        $alternatifs = Alternatif::get();
        $bobots = Bobot::with('crips')->get();
      // return $evaluasis;
      return view('pages.admin.evaluasi.index', compact('evaluasis', 'alternatifs', 'bobots'));
   }  }
```

4. MatrikController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Alternatif;
use App\Models\Bobot;
use App\Models\Crips;
use App\Models\Evaluasi;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\App;

class MatrikController extends Controller
{

    // tampil data
    public function index()
    {
        $bobots = Bobot::query()->select('id', 'symbol')->get();
        $alternatifs = Alternatif::query()
            ->get()
            ->map(function ($collect) {
                $collect['table'] = App::call(MatrikPenetapan::class, ['request' => ['id' => $collect->id, 'year' => date('Y'), 'month' => date('m')]]);
                return $collect;
            });
        // return $alternatifs;
        $alternatif = Alternatif::query()
            ->get()
            ->map(function ($collect) {
                $collect['table'] = Ap
        p::call(Perhitungan::class, ['request' => ['id' => $collect->id, 'year' => date('Y'), 'month' => date('m')]]);
                $total = $collect['table']->pluck('nilai_perferensi')->sum();
                    $collect['total_nilai'] = $total;
                    return $collect;  });
                return view('pages.admin.matrik.index', compact('bobots', 'alternatifs', 'alternatif'));   }
// function cari
    public function cari(Request $request){
                $bobots  =  Bobot::query()->select('id', 'symbol')->get();
        $alternatifs = Alternatif::query()
            ->get()
            ->map(function ($collect) use($request) {
                $collect['table'] = App::call(MatrikPenetapan::class,  ['request' => ['id' => $collect->id, 'year' => $request->year, 'month' => $request->month]]);
                return $collect;
            });
        // return $alternatifs;
        $alternatif =  Alternatif::query()
            ->get()
            ->map(function ($collect) use($request) {
                $collect['table'] = App::call(Perhitungan::class, ['request' => ['id' => $collect->id,  'year'  => $request->year, 'month' => $request->month]]);
                $total = $collect['table']->pluck('nilai_perferensi')->sum();
                    $collect['total_nilai'] = $total;
                return $collect;
            });
        return view('pages.admin.matrik.index', compact('bobots', 'alternatifs', 'alternatif'));
    }
}
```

5. PerankinganController.php

```
<?php
namespace App\Http\Controllers;

use App\Models\Alternatif;
use App\Models\Bobot;
use Illuminate\Http\Request;
use Carbon\Carbon;
use Illuminate\Support\Facades\App;

class PerankinganController extends
Controller
{
    // tampil data
    public function index()
    {
        $bobots = Bobot::get();
        $alternatifs = Alternatif::query()
            ->get()
            ->map(function ($collect) {
                $collect['table'] = App::call
(Perhitungan::class, ['request' => ['id'
                =>                  $collect-
>id, 'year' => date('Y'), 'month' => da
te('m')]]);
                $total = $collect['table']-
>pluck('nilai_perferensi')->sum();
                $collect['total_nilai'] = $total;
                return $collect;
            })
            ->sortByDesc('total_nilai');
        $year = $request['year'] ?? date('
Y');
        $month = $request['month'] ?? d
ate('m');
                $m = Carbon::now()-
        >month($month)->locale('id')-
        >translatedFormat('F');
```

```
        $date = $m . ' ' .$year;
        // return $alternatifs;
         return view('pages.admin.
perangkingan.index', compact('
bobots', 'alternatifs','date'));
    }
// function pencarian
    public function cari(Request
$request)
    {
        // return $request;
        $bobots = Bobot::get();
         $alternatifs = Alternatif::q
uery()
            ->get()
                                    -
>map(function ($collect) use ($reque
st) {
                $collect['table'] = App::call
(Perhitungan::class, ['request' => ['id'
 => $collect->id, 'year' => $request-
>year,    'month'    =>    $request-
>month]]);
                $total = $collect['table']-
>pluck('nilai_perferensi')->sum();
                $collect['total_nilai'] = $total;
                return $collect;
            })
            ->sortByDesc('total_nilai');
            $year = $request['year'] ?? dat
e('Y');
```

```
            $month = $request['month'] ?
? date('m');
                    $m  =  Carbon::now()-
>month($month)->locale('id')-
>translatedFormat('F');
          $date = $m . ' ' .$year;
      // return $alternatifs;
         return view('pages.admin.peran
gkingan.index', compact('bobots', 'alt
ernatifs','date'));
   } // untuk cetak table
   public function cetak(Request $req
uest) {   $bobots = Bobot::get();
      $alternatifs = Alternatif::query()
          ->get()
          >map(function ($collect)use($
request) {
              $year = $request['year'] ??
date('Y');    $month = $request['mont
h'] ?? date('m');
                     $collect['table'] = Ap
      p::call(Perhitungan::class, ['req
      uest'    =>    ['id'    =>    $collect-
      >id, 'year' => $year, 'month' =>
      $month]]);
               $total = $collect['table']-
      >pluck('nilai_perferensi')-
      >sum();
                  $collect['total_nilai']
      = $total;
                 return $collect;
                  })                -
>sortByDesc('total_nilai');
      $num = 1; $year = $reques
t['year'] ?? date('Y');
      $month = $request['month'
] ?? date('m');
           $m  =  Carbon::now()-
>month($month)->locale('id')-
>translatedFormat('F');
      $date = $m . ' ' .$year;
        return view('pages.admin.
perangkingan.cetak',   compact('
bobots', 'alternatifs', 'num','date'
));
   } }
```