

## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Berdasarkan hasil yang didapat dalam penulisan ini, maka diperoleh kesimpulan sebagai berikut :

1. *Web e-commerce* yang dibuat dapat membantu pelanggan dalam mempercepat pemilihan produk plastik yang sesuai dengan kriteria dan keinginan.
2. *Web e-commerce* yang dibuat dapat Membantu UD. Aneka Plastik dalam pemasaran produk plastik ke wilayah yang lebih luas.

#### 5.2 Saran

Saran yang di rekomendasikan penulis untuk pengembangan sistem ini lebih lanjut diantaranya :

1. Diberikannya pilihan dalaman pembayaran tidak hanya transfer saja tapi juga bisa menggunakan *credit*.
3. Perlu adanya penambahan fitur dan fungsi fasilitas lain, sehingga pelanggan akan terus mengunjungi *web* dan meningkatkan kemungkinan berbelanja.

## DAFTAR PUSTAKA

- Abdullah, T., & Tantri, F. (2016). *Manajemen Pemasaran*. Depok: PT Raja Grafindo Persada.
- Afrianto, I., Heryandi, A., Finadhita, A., & Atin, S. (2021). User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program. *International Journal of Information System & Technology*, 5(3).
- Ahmad, A. S. (2013). *Modifikasi CMS Template Lokomedia*. Yogyakarta: Penerbit Garudhawaca.
- Anwar, A. S., Utomo, A. P., & Nugraha, F. (2018). Sistem Informasi Produksi Plastik Pada UD. Bagas Mulya Mejobo Kudus Berbasis Web. *Jurnal Sitech*, 1(1), 49–56.
- Aziz, A., Setiawan, I., Krisbiantoro, D., Riyanto, & Setiawan, F. D. (2019). *Panduan Pemilu Desa Berbasis Website*. Sleman: Deepublish.
- Candra, H. K. (2019). *Pengantar Teknologi Informasi*. Banjarmasin: Poliban Press.
- Dadan, & Developers, K. (2015). *Membuat CMS Multifitur*. Jakarta: Elex Media Komputindo.
- Djahir, Y., & Pratita, D. (2015). *Bahan Ajar Sistem Informasi Manajemen*. Yogyakarta: Deepublish.
- Enterprise, J. (2015). *Pengenalan Visual Studio 2013*. Jakarta: PT Elex Media Komputindo.
- Hendrastianto, Z. A. (2022). Plastik: Pengertian, Sejarah, Jenis, Proses Pembuatan, dan Bahan Baku. Retrieved from <https://foresteract.com/plastik/>
- Hikmah, A. B., Supriadi, D., & Alawiyah, T. (2015). *Cara Cepat Membangun Website dari Nol Studi Kasus : Web Dealer Motor*. Yogyakarta: CV Andi Offset.
- Malau, H. (2016). *Manajemen Pemasaran Teori dan Aplikasi Pemasaran Era Tradisional Sampai Era Modernisasi Global*. Bandung: Alfabeta.
- Mulyadi. (2016). *Sistem Informasi Akuntansi*. Jakarta: Salemba Empat.
- Muslihudin, M., & Oktafianto. (2016). *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Andi Offset.
- Nur, M. (2019). *Situs situs pengunduh video Youtube dan Mp3 Terbaik*. Jakarta: MiftaChun.
- Pane, S. F., Zamzam, M., & Fadillah, M. D. (2020). *Membangun Aplikasi Peminjaman*

- Jurnal Menggunakan Aplikasi Oracle Apex Online*. Bandung: Kreatif Industri Nusantara.
- Patma, T. S., Maskan, M., & Utaminingsih, A. (2018). *Sistem Informasi Manajemen Guna Mendukung Keputusan*. Malang: UPT Percetakan dan Penerbitan Polinema.
- Pinto, V. H. S. C., Oliveira, R. R., Vilela, R. F., & Souza, S. R. S. (2019). Evaluating the User Acceptance Testing for Multi-tenant Cloud Applications. *8th International Conference on Cloud Computing and Services Science*, 47–56.
- Prima, A. M., & Mulyana, D. I. (2021). Implementasi System Penjualan Barang Plastik Berbasis Web Dengan Metode Devops Pada PT. Cahaya Perdana Plastic Jakarta Utara. *Jurnal Sosial Dan Teknologi*, 1(7), 581–588.
- Purbadian, Y. (2016). *Trik Cepat Membangun Aplikasi Berbasis Web dengan Framework CodeIgniter*. Yogyakarta: Andi Offset.
- Sidik, A., Waluyo, E. T. B., & Susilawati, S. (2018). Perancangan Sistem Informasi Laporan Persediaan Barang Jadi PT Duta Prima Plasindo. *Sisfotek Global*, 8(1), 7–12.
- Sujarweni, V. W. (2015). *Statistik untuk Bisnis dan Ekonomi*. Yogyakarta: Pustaka Baru Press.
- Sukamto, R. A., & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Sutabri, T. (2016). *Sistem Informasi Manajemen*. Yogyakarta: CV. Andi Offset.
- Sutra, L., & Mujiastuti, R. (2021). Penjualan Masterbatch (PEWARNA PLASTIK) Berbasis E-Commerce Web Menggunakan Metode Hybriid. *Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, 11(2), 8–17.
- Wicaksono, R. S. (2017). *Rekayasa Perangkat Lunak*. Malang: Seribu Bintang.
- Yanto, R. (2016). *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta: Deepublish.
- Yudhanto, Y., & Prasetyo, H. A. (2019). *Mudah Menguasai Framework laravel*. Jakarta: PT Elex Media Komputindo.

## DAFTAR RIWAYAT HIDUP



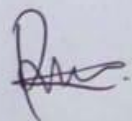
### Data Pribadi

Nama Lengkap : Ronal  
Tempat/Tanggal Lahir : Tangerang, 12 Juli 2000  
Jenis Kelamin : Laki-laki  
Alamat : Jl. Cukanggalih Desa. Ciakar Kec. Panongan  
Agama : Buddha  
Telepon : 0822-4689-3474  
Email : Ronalsendi@gmail.com

### Pendidikan Formal

2006 – 2012 : SDN Ranca Kelapa II  
2012 – 2015 : SMP Yupentek 1 Legok  
2015 – 2018 : SMK Lentera Harapan  
2018 – 2022 : Universitas Buddhi Dharma (Sistem Informasi)

Tangerang, 26 Januari 2023



**Ronal**

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENJUALAN PLASTIK  
BERBASIS *ONLINE* DENGAN METODE PENGUJIAN *USER ACCEPTANCE*  
*TESTING***

Nama : Ronal  
NIM : 20180700037  
Universitas : Universitas Buddhi Dharma

**REQUIREMENT ELICITION**

NO	Analisa Kebutuhan	Keterangan
	Saya Ingin Aplikasi Ini dapat :	
	Adanya Halaman Login	
	Adanya Halaman checkout Barang	
	Adanya Halaman user	
	Adanya Halaman history order	
	Adanya Halaman Registrasi	

Tangerang, 9. Desember 2022


Pembimbing,



Rudy Arijanto, S.Kom., M.Kom.

NIDN. 0415077105

Responden,



Eugun

Mahasiswa,



Ronal.

20180700037

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENJUALAN PLASTIK  
BERBASIS *ONLINE* DENGAN METODE PENGUJIAN *USER ACCEPTANCE*  
*TESTING***

Nama : Ronal  
NIM : 20180700037  
Universitas : Universitas Buddhi Dharma

**REQUIREMENT ELICITION**

NO	Analisa Kebutuhan	Keterangan
	Saya Ingin Aplikasi Ini dapat :	
	Adanya Halaman Dashboard	
	Adanya Halaman Payment	
	Adanya Halaman List Category	
	Adanya Halaman List Product	
	Adanya Halaman Shop	

Tangerang, 9. Desember 2022

Pembimbing,



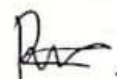
**Rudy Arijanto, S.Kom., M.Kom.**

**NIDN. 0415077105**

Responden,



Mahasiswa,



**Ronal.**

**20180700037**

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENJUALAN PLASTIK  
BERBASIS *ONLINE* DENGAN METODE PENGUJIAN *USER ACCEPTANCE*  
*TESTING***

Nama : Ronal  
NIM : 20180700037  
Universitas : Universitas Buddhi Dharma

**REQUIREMENT ELICITION**

NO	Analisa Kebutuhan	Keterangan
	Saya Ingin Aplikasi Ini dapat :	
	Adanya Halaman List Stock Barang	
	Adanya Halaman List order	
	Adanya Halaman order Reject	
	Adanya Halaman order finish	
	Adanya Halaman Logout.	

Tangerang, 9 Desember 2022


Pembimbing,



**Rudy Ariyanto, S.Kom., M.Kom.**

**NIDN. 0415077105**

Responden,



Darmadi

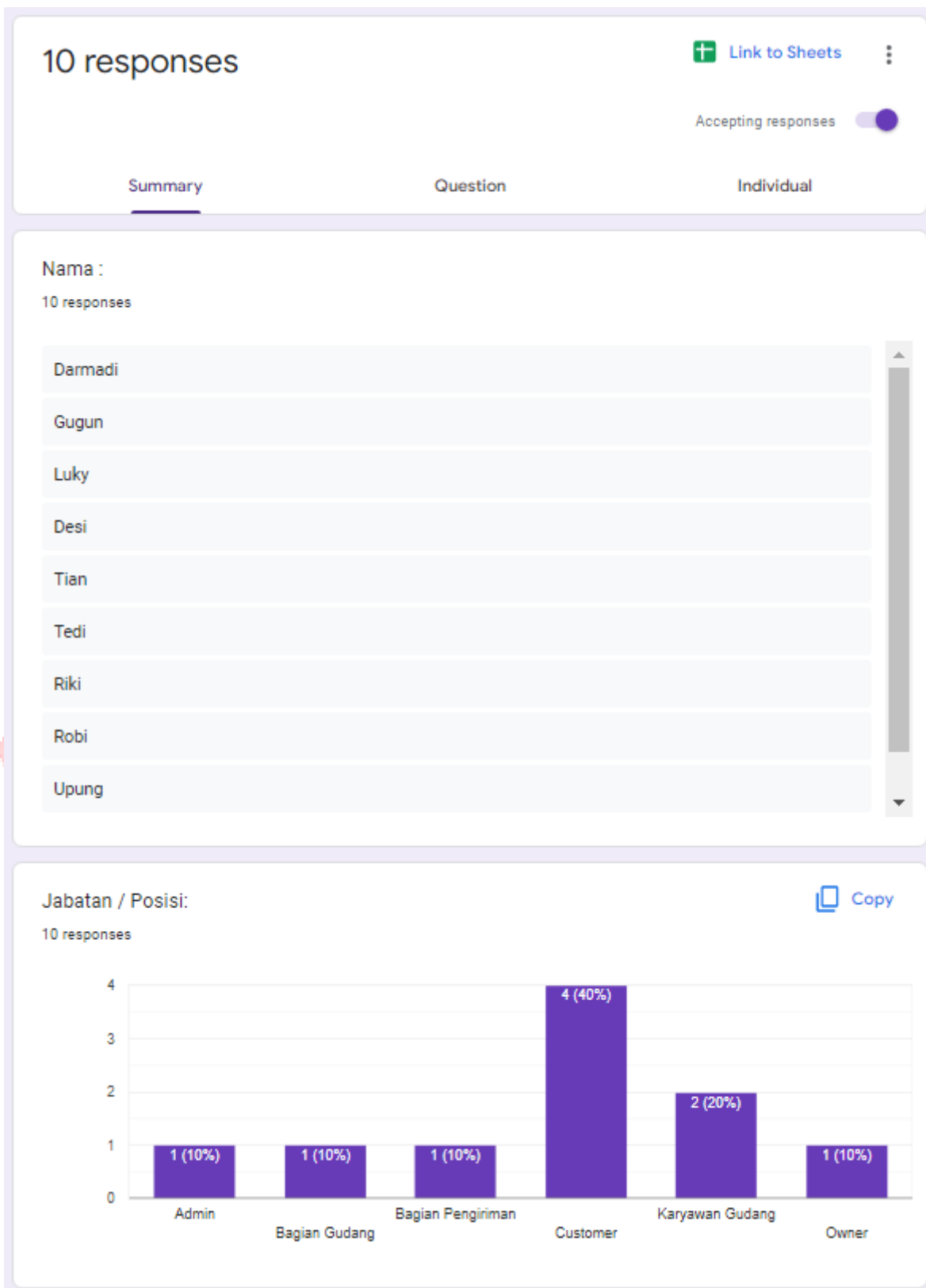
Mahasiswa,



**Ronal.**

**20180700037**

# KUESIONER GOOGLE FORM

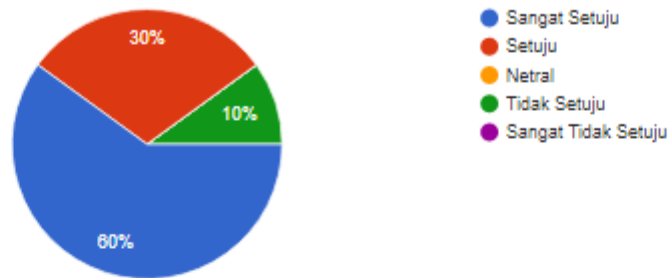




1. Apakah aplikasi mudah dioperasikan ?

 Copy

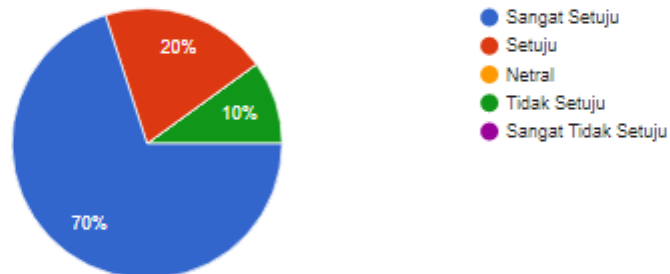
10 responses



2. Apakah aplikasi ini mudah digunakan ?

 Copy

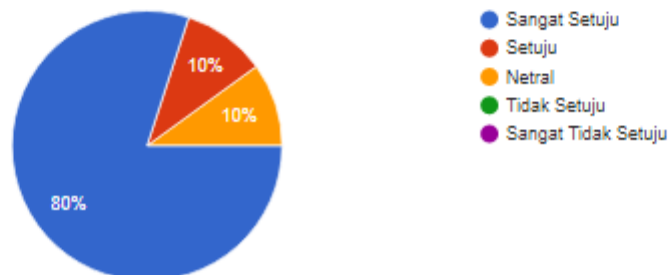
10 responses



3. Apakah aplikasi bermanfaat bagi pengguna ?

 Copy

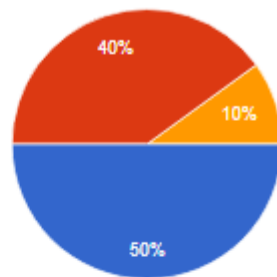
10 responses



4. Apakah aplikasi ini sesuai dengan kebutuhan ?

 Copy

10 responses

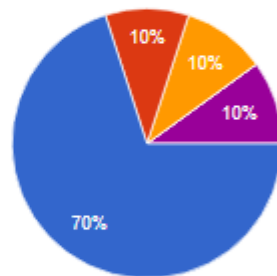


- Sangat Setuju
- Setuju
- Netral
- Tidak Setuju
- Sangat Tidak Setuju

5. Apakah aplikasi nyaman digunakan ?

 Copy

10 responses

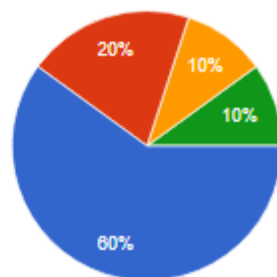


- Sangat Setuju
- Setuju
- Netral
- Tidak Setuju
- Sangat Tidak Setuju

6. Apakah tata letak dari aplikasi ini sudah sesuai?

 Copy

10 responses

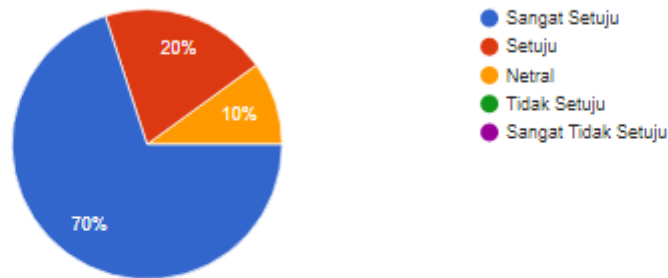


- Sangat Setuju
- Setuju
- Netral
- Tidak Setuju
- Sangat Tidak Setuju

7. Apakah menu dalam aplikasi sudah cukup lengkap ?

 Copy

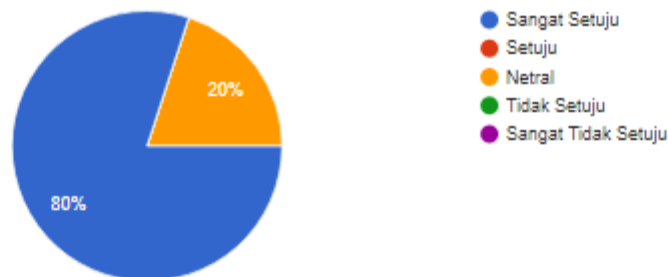
10 responses



8. Apakah hasil yang ditampilkan aplikasi sesuai dengan kebutuhan / keinginan Anda ?

 Copy

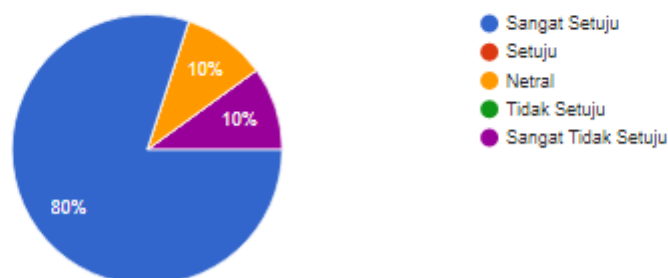
10 responses



9. Apakah informasi yang disediakan oleh aplikasi mudah di mengerti ?

 Copy

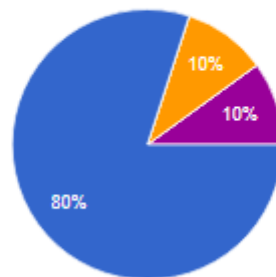
10 responses



10. Secara keseluruhan apakah penggunaan aplikasi ini memuaskan ?

 Copy

10 responses



- Sangat Setuju
- Setuju
- Netral
- Tidak Setuju
- Sangat Tidak Setuju





UNIVERSITAS BUDDHI DHARMA

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang  
021 5517853 / 021 5586822 admin@buddhidharma.ac.id

## KARTU BIMBINGAN TA/SKRIPSI

NIM : 20180700037  
Nama Mahasiswa : RONAL  
Fakultas : Sains dan Teknologi  
Program Studi : Sistem Informasi  
Jenjang : Strata Satu  
Tahun Akademik/Semester : 2022/2023 Ganjil  
Dosen Pembimbing : Rudy Arijanto, S.Kom.,M.Kom.  
Judul Skripsi : ANALISIS DAN PERANCANGAN SISTEM INFORMASI  
PENJUALAN PLASTIK BERBASIS ONLINE DENGAN METODE  
PENGUJIAN USER ACCEPTANCE TESTING

Tanggal	Catatan	Paraf
2022-09-06	Konsultasi judul dan persetujuan judul skripsi	
2022-09-20	BAB I. Pendahuluan - Latar Belakang - Tujuan dan Manfaat	
2022-10-04	BAB II. Landasan Teoritis - Tambahkan Teori - Buat Daftar Pustaka	
2022-10-18	BAB III. Sistem Yang Berjalan - Struktur Organisasi - Activity Diagram	
2022-11-01	BAB IV. Sistem Yang Diusulkan - Activity Diagram - Use Case Diagram	
2022-11-15	BAB IV. Sistem Yang Diusulkan - ERD - Sequence Diagram	
2022-11-29	BAB IV. Sistem Yang Diusulkan - Spesifikasi File - Rancangan Input/Output	
2022-12-06	BAB V. Simpulan dan Saran - Simpulan - Saran	
2022-12-13	- Pemeriksaan seluruh skripsi - Persetujuan Jilid	

Mengetahui  
Ketua Program Studi

Benny Daniawan, M.Kom

Tangerang, 09 January 2023

Pembimbing

Rudy Arijanto, S.Kom.,M.Kom.



# UD. ANEKA PLASTIK

Jl. Jalan Raya Ranca Iyuh, Simpang 3 Pasar Korelet Desa Ranca Iyuh  
Kecamatan Panongan Kab. Tangerang – Banten 15710  
Telp. (081) 280285307

---

## SURAT KETERANGAN RISET

Yang bertanda tangan dibawah ini menerangkan bahwa:

<b>Nama</b>	<b>: Ronal</b>
<b>NIM</b>	<b>: 20180700037</b>
<b>Mahasiswa dari</b>	<b>: Universitas Buddhi Dharma</b>
<b>Program Studi</b>	<b>: Sistem Informasi</b>
<b>Fakultas</b>	<b>: Sains &amp; Teknologi</b>

Adalah benar telah melakukan penelitian di UD. ANEKA PLASTIK, sejak tanggal 5 September 2022 sampai dengan tanggal 9 Desember 2022.

Demikian surat keterangan ini agar dapat di pergunakan sebagaimana mestinya.

Tangerang, 9 Desember 2022

Hormat Kami



Owner



**PEMERINTAH KABUPATEN TANGERANG**  
**DESA RANCA IYUH**  
**KECAMATAN PANONGAN**

Alamat : Jl. Ranca Iyuh No. 1 Kec. Panongan - Tangerang

**SURAT KETERANGAN**  
**Nomor : 563/ Ds,Rci / VIII / 2021**

Yang bertanda tangan dibawah ini Kepala Desa Rancaiyuh Kecamatan Panongan Kabupaten Tangerang, dengan ini menerangkan bahwa :

Nama : **DARMADI**  
Jenis Kelamin : Laki-Laki  
Tempat tanggal lahir : Tangerang, 10-03-1984  
Kewarganegaraan : Indonesia  
Agama : Khatolik  
Pekerjaan : Karyawan Swasta  
Alamat : Kp. Kolelet Rt. 002/004 Desa Ranca Kalapa  
: Kec. Panongan Kab. Tangerang

Benar nama orang tersebut diatas pada saat ini memiliki usaha sebagai berikut :

Nama Usaha : **ANEKA PLASTIK**  
Jenis Usaha : **"MENJUAL ANEKA PLASTIK"**  
Alamat Usaha : Jl. Raya Pasar Korelet Rt. 011/003 Desa Rancaiyuh  
: Kec. Panongan Kab. Tangerang

Demikian surat keterangan usaha ini dibuat dengan sebenarnya, kepada yang berkepentingan agar mengetahuinya.

Rancaiyuh, 05 Agustus 2021  
An. Kepala Desa Rancaiyuh  
Kaur/Umum  
SEKRETARIS DESA  
RANCAIYUH  
**ROBISUGARA**

## LISTING PROGRAM

### OrderPaymentController.php

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

use App\Models\Order;
use App\Models\OrderPayment;
use Illuminate\Http\Request;

class OrderPaymentController extends Controller
{
    public function index(Request $request)
    {
        if ($request->currentuser) {
            return Order::where('status', '=', 0)->where('customer_id', '=', $request->currentuser['id']->first();
        }
    }
    public function store(Request $request)
    {
        $order = Order::doesntHave('orderPayment')->where('customer_id', '=', $request->currentuser)->first();
        $photo = $request->file('image')->getClientOriginalName();
        $photoName = date('dmy') . $request->name . uniqid() . $photo;
        $product = OrderPayment::create([
            'order_id' => $order->id,
            'amount' => $request->total,
            'status' => 1,
            'photo' => $photoName
        ]);
        $request->image->move(public_path('storage/images'), $photoName);
        $order->status = 1;
        $order->save();
        return true;
    }
}
```



## ProductController.php

```
<?php

namespace App\Http\Controllers;

use Inertia\Inertia;
use App\Models\Product;
use App\Models\Category;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Redirect;

class ProductController extends Controller
{
    private $header;
    private $currentNav = "";
    public function filterMenu()
    {
        $menus = [];
        foreach (config('setting.menu') as $menu) {
            if ($menu['text'] == 'User') {
                if (Auth::user()->is_admin === 1) {
                    $menus[] = $menu;
                }
            } else {
                $menus[] = $menu;
            }
        }
        return $menus;
    }
    public function returnProps(array $props = [])
    {
        return array_merge([
            'header' => $this->header,
            'menu' => [
                'currentSubMenu' => $this->header,
                'currentNav' => $this->currentNav,
                'list' => $this->filterMenu(),
            ]
        ], $props);
    }
}
```

```

    ], $props);
}
public function index()
{
    $data = Product::with('category', 'stock')->has('category')->orderBy('id', 'desc')->paginate(10);
    $this->header = 'List Product';
    $this->currentNav = 'Product & Stock';
    foreach ($data as &$value) {
        $value->categories = implode(',', array_column($value->category->toArray(), 'name'));
        $value->stockProduct = empty($value->stock->quantity) ? 0 : $value->stock->quantity;
        unset($value);
    }
    return Inertia::render('Report', $this->returnProps([
        'table' => [
            'title' => 'List Product',
            'pageRoute' => 'product.index',
            'action' => 'product',
            'button' => true,
            'list' => $data,
            'header' => [
                [
                    'Header' => '#Id',
                    'accessor' => 'id',
                ],
                [
                    'Header' => 'Product Name',
                    'accessor' => 'name',
                ],
                [
                    'Header' => 'Price',
                    'accessor' => 'price',
                    'format' => 'currency',
                    'prefix' => 'IDR.'
                ],
                [
                    'Header' => 'unit',
                    'accessor' => 'unit',
                ],
                [
                    'Header' => 'Category',

```

```

        'accessor' => 'categories',
    ],
    [
        'Header' => 'Stock',
        'accessor' => 'stockProduct',
    ],
    [
        'Header' => 'Image',
        'accessor' => 'photo',
        'format' => 'photo'
    ],
    [
        'Header' => 'Created At',
        'accessor' => 'created_at',
    ],
    [
        'Header' => 'Updated At',
        'accessor' => 'updated_at',
    ],
    [
        'Header' => 'Delete',
        'accessor' => 'id',
        'format' => 'action:delete',
    ],
    [
        'Header' => 'Edit',
        'accessor' => 'id',
        'format' => 'action:edit',
    ],
    ]
    ]
    ));
}

public function edit(Product $product)
{
    $this->currentNav = 'Product & Stock';
    $this->header = 'Edit Product';
    $Category = Category::get();
    $selection = [];
    foreach ($Category as $value) {

```

```

$selection[] = [
    'label' => $value->name,
    'value' => $value->id
];
}
$defaultValue = [];
foreach ($product->category as $default) {
    $defaultValue[] = [
        'label' => $default->name,
        'value' => $default->id
    ];
}
return Inertia::render('Form', $this->returnProps([
    'form' => [
        'title' => 'Edit Product ' . $product->name,
        'action' => route('product.update', $product),
        'type' => 'update',
        'design' => [
            [
                'type' => 'text',
                'label' => 'Product Name',
                'name' => 'name',
                'require' => true,
                'value' => $product->name
            ],
            [
                'type' => 'currency',
                'label' => 'Product Price',
                'name' => 'price',
                'require' => true,
                'value' => $product->price
            ],
            [
                'type' => 'text',
                'label' => 'Unit',
                'name' => 'unit',
                'require' => true,
                'value' => $product->unit
            ],
            [

```

```

        'type' => 'select',
        'label' => 'Product Category',
        'name' => 'category',
        'require' => false,
        'value' => $defaultValue,
        'custom' => [
            'options' => $selection,
            'isMulti' => true,
        ]
    ],
    [
        'type' => 'photo',
        'label' => 'Product Image',
        'name' => 'image',
        'require' => true,
        'value' => $product->photo
    ]
]
]);
}
public function update(Request $request, Product $product)
{
    $this->validate($request, [
        'name' => "required|unique:products,name,{ $product->name },name",
        'image' => 'mimes:png,jpg,jpeg,JPG',
        'price' => 'required|integer',
        'category' => 'required',
        'unit' => 'required'
    ]);

    $product->name = $request->name;
    $product->price = $request->price;
    $product->unit = $request->unit;
    $product->category()->detach();
    foreach ($request->category as $value) {
        $product->category()->attach($value['value']);
    }
    if ($request->image) {
        $productPhoto = $request->file('image')->getClientOriginalName();
    }
}

```

```

    $photoName = date('dmy') . $request->name . uniqid() . $productPhoto;
    $product->photo = $photoName;
    $request->image->move(public_path('storage/images'), $photoName);
}
$product->save();
return redirect()->route('product.index');
}
public function create()
{
    $this->currentNav = 'Product & Stock';
    $this->header = 'List Product';
    $Category = Category::get();
    $selection = [];
    foreach ($Category as $value) {
        $selection[] = [
            'label' => $value->name,
            'value' => $value->id
        ];
    }
    return Inertia::render('Form', $this->returnProps([
        'form' => [
            'title' => 'Create new Product',
            'action' => route('product.store'),
            'design' => [
                [
                    'type' => 'text',
                    'label' => 'Product Name',
                    'name' => 'name',
                    'require' => true,
                ],
                [
                    'type' => 'currency',
                    'label' => 'Product Price',
                    'name' => 'price',
                    'require' => true,
                ],
                [
                    'type' => 'text',
                    'label' => 'Unit',
                    'name' => 'unit',
                ]
            ]
        ]
    ]));
}

```

```

        'require' => true,
    ],
    [
        'type' => 'select',
        'label' => 'Product Category',
        'name' => 'category',
        'require' => false,
        'custom' => [
            'options' => $selection,
            'isMulti' => true,
        ]
    ],
    [
        'type' => 'photo',
        'label' => 'Product Image',
        'name' => 'image',
        'require' => true,
    ]
    ]
    ]
    ));
}
public function destroy(Request $request, Product $product)
{
    $product->delete();
    return Redirect::route('product.index');
}
public function store(Request $request)
{
    $this->validate($request, [
        'name' => 'required|unique:products,name',
        'image' => 'required|mimes:png,jpg,jpeg,JPG',
        'price' => 'required|integer',
        'category' => 'required',
        'unit' => 'required'
    ]);
    $productPhoto = $request->file('image')->getClientOriginalName();
    $photoName = date('dmy') . $request->name . uniqid() . $productPhoto;
    $product = Product::create([
        'name' => $request->name,

```

```

        'price' => $request->price,
        'unit' => $request->unit,
        'photo' => $photoName
    ]);
    foreach ($request->category as $value) {
        $product->category()->attach($value['value']);
    }
    $request->image->move(public_path('storage/images'), $photoName);
    return redirect()->route('product.index');
}
}

```

### StockController.php

```

<?php
namespace App\Http\Controllers;

use Inertia\Inertia;
use App\Models\Stock;
use App\Models\Product;
use Illuminate\Http\Request;
use App\Rules\CheckListProductExist;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Redirect;

class StockController extends Controller
{
    private $currentNav = "";
    public function filterMenu()
    {
        $menus = [];
        foreach (config('setting.menu') as $menu) {
            if ($menu['text'] == 'User') {
                if (Auth::user()->is_admin === 1) {
                    $menus[] = $menu;
                }
            } else {
                $menus[] = $menu;
            }
        }
    }
}

```



```

    }
    return $menus;
}
public function returnProps(array $props = [])
{
    return array_merge([
        'header' => $this->header,
        'menu' => [
            'currentSubMenu' => $this->header,
            'currentNav' => $this->currentNav,
            'list' => $this->filterMenu()
        ]
    ], $props);
}
public function index()
{
    $data = Stock::with('product')->has('product')->orderBy('id', 'desc')->paginate(10);
    foreach ($data as &$value) {
        $value->productName = $value->product->name;
        unset($value);
    }
    $this->header = 'Update Stock';
    $this->currentNav = 'Product & Stock';
    return Inertia::render('Report', $this->returnProps([
        'table' => [
            'title' => 'List Stock',
            'pageRoute' => 'stock.index',
            'action' => 'stock',
            'list' => $data,
            'header' => [
                [
                    'Header' => '#Id',
                    'accessor' => 'id',
                ],
                [
                    'Header' => 'Product name',
                    'accessor' => 'productName',
                ],
                [
                    'Header' => 'Product Id',

```

```

        'accessor' => 'product_id',
    ],
    [
        'Header' => 'Quantity',
        'accessor' => 'quantity',
    ],
    [
        'Header' => 'Created At',
        'accessor' => 'created_at',
    ],
    [
        'Header' => 'Updated At',
        'accessor' => 'updated_at',
    ],
    [
        'Header' => 'Delete',
        'accessor' => 'id',
        'format' => 'action:delete',
    ],
    [
        'Header' => 'Edit',
        'accessor' => 'id',
        'format' => 'action:edit',
    ],
    ]
    ]
    ));
}
public function create()
{
    $this->currentNav = 'Product & Stock';
    $this->header = 'Update Stock';
    $product = Product::doesntHave('stock')->has('category')->get();
    $selection = [];
    foreach ($product as $value) {
        $selection[] = [
            'label' => $value->name,
            'value' => $value->id
        ];
    }
}

```

```

return Inertia::render('Form', $this->returnProps([
    'form' => [
        'title' => 'Update Stock',
        'action' => route('stock.store'),
        'design' => [
            [
                'type' => 'select',
                'label' => 'Product',
                'name' => 'product',
                'require' => false,
                'custom' => [
                    'options' => $selection,
                    'isMulti' => false,
                ]
            ],
            [
                'type' => 'number',
                'label' => 'Quantity',
                'name' => 'qty',
                'require' => true,
            ],
        ]
    ]
));
}
public function store(Request $request)
{
    $this->validate($request, [
        'qty' => 'required|integer',
        'product' => ['required', new CheckListProductExist()]
    ]);
    Stock::create([
        'quantity' => $request->qty,
        'product_id' => $request->product['value'],
    ]);
    return redirect()->route('stock.index');
}
public function update(Request $request, Stock $stock)
{
    $this->validate($request, [

```

```

        'qty' => 'required|integer',
    ]);
    $stock->quantity = $request->qty;
    $stock->save();
    return redirect()->route('stock.index');
}

public function edit(Stock $stock)
{
    $this->currentNav = 'Product & Stock';
    $this->header = 'Update Stock';
    $product = Product::doesntHave('stock')->get();
    foreach ($product as $value) {
        $selection[] = [
            'label' => $value->name,
            'value' => $value->id
        ];
    }
    $defaultValue[] = [
        'label' => $stock->product->name,
        'value' => $stock->product_id
    ];
    return Inertia::render('Form', $this->returnProps([
        'form' => [
            'title' => 'Edit stock Product ' . $stock->product->name,
            'action' => route('stock.update', $stock),
            'type' => 'update',
            'design' => [
                [
                    'type' => 'number',
                    'label' => 'Quantity',
                    'name' => 'qty',
                    'value' => $stock->quantity,
                    'require' => true,
                ],
            ]
        ]
    ]));
}

public function destroy(Stock $stock)
{

```

```

        $stock->delete();
        return Redirect::route('stock.index');
    }
}

```

### AuthenticatedSessionController.php

```

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

use App\Http\Controllers\Controller;
use App\Http\Requests\Auth>LoginRequest;
use App\Providers\RouteServiceProvider;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Route;
use Inertia\Inertia;

class AuthenticatedSessionController extends Controller
{
    /**
     * Display the login view.
     *
     * @return \Inertia\Response
     */
    public function create()
    {
        return Inertia::render('Auth/Login', [
            'canResetPassword' => Route::has('password.request'),
            'status' => session('status'),
        ]);
    }

    /**
     * Handle an incoming authentication request.
     *
     * @param \App\Http\Requests\Auth>LoginRequest $request
     * @return \Illuminate\Http\RedirectResponse
     */
}

```

```

public function store(LoginRequest $request)
{
    $request->authenticate();

    if (Request::capture()->expectsJson()) {
        $token = $request->user()->createToken($request->password . $request->email);
        $response = [
            'status' => 'success',
            'msg' => 'Login successfully',
            'content' => [
                'status_code' => 200,
                'access_token' => $token->plainTextToken,
                'token_type' => 'Bearer',
                'user_name' => auth()->user()->name,
                'user_email' => auth()->user()->email,
                'user_id' => auth()->user()->customer->id,
            ]
        ];
        return response()->json($response, 200);
    } else {
        $request->session()->regenerate();
        return redirect()->intended(RouteServiceProvider::HOME);
    }
}

/**
 * Destroy an authenticated session.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\RedirectResponse
 */
public function destroy(Request $request)
{
    Auth::guard('web')->logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();
}

```

```

        return redirect('/');
    }
}

```

## NewPasswordController.php

```

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

use App\Http\Controllers\Controller;
use Illuminate\Auth\Events>PasswordReset;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades>Password;
use Illuminate\Support\Str;
use Illuminate\Validation\Rules;
use Illuminate\Validation\ValidationException;
use Inertia\Inertia;

class NewPasswordController extends Controller
{
    /**
     * Display the password reset view.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\View\View
     */
    public function create(Request $request)
    {
        return Inertia::render('Auth/ResetPassword', [
            'email' => $request->email,
            'token' => $request->route('token'),
        ]);
    }

    /**
     * Handle an incoming new password request.
     *
     * @param \Illuminate\Http\Request $request

```

```

* @return \Illuminate\Http\RedirectResponse
*
* @throws \Illuminate\Validation\ValidationException
*/
public function store(Request $request)
{
    $request->validate([
        'token' => 'required',
        'email' => 'required|email',
        'password' => ['required', 'confirmed', Rules\Password::defaults()],
    ]);

    // Here we will attempt to reset the user's password. If it is successful we
    // will update the password on an actual user model and persist it to the
    // database. Otherwise we will parse the error and return the response.
    $status = Password::reset(
        $request->only('email', 'password', 'password_confirmation', 'token'),
        function ($user) use ($request) {
            $user->forceFill([
                'password' => Hash::make($request->password),
                'remember_token' => Str::random(60),
            ]->save();

            event(new PasswordReset($user));
        }
    );

    // If the password was successfully reset, we will redirect the user back to
    // the application's home authenticated view. If there is an error we can
    // redirect them back to where they came from with their error message.
    if ($status == Password::PASSWORD_RESET) {
        return redirect()->route('login')->with('status', __($status));
    }

    throw ValidationException::withMessages([
        'email' => [trans($status)],
    ]);
}
}

```