

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

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

1. Dengan adanya sistem informasi *e-commerce* berbasis *web* ini maka dapat lebih mempermudah pelanggan melakukan pemesanan produk secara *online*.
2. Diharapkan dengan adanya penerapan sistem informasi *e-commerce* berbasis *web* pada PT. Qiao Xing Industry dapat bersaing dalam pemasaran produk untuk meningkatkan omset penjualan perusahaan.
3. Hasil uji sistem dengan metode *Technology Acceptance Model* dapat disimpulkan dalam kriteria sangat setuju dengan nilai 89,54%. dengan hasil variabel *perceived usefulness* sebesar 86,86%, hasil variabel *perceived easy of use* sebesar 88,4%, hasil variabel *usability* sebesar 90,2%, hasil variabel *graphic design* sebesar 91,2%, hasil variabel *compatibility* sebesar 89,3% dan hasil variabel *functionality* sebesar 91,3%.

5.2 Saran

Ada pula hasil yang didapat dalam penulisan skripsi ini, maka diperoleh saran untuk pengembangan aplikasi agar lebih baik adalah sebagai berikut :

1. Sistem informasi yang telah dibangun ini agar dijadikan bahan untuk pengembangan sistem lebih lanjut sehingga seluruh transaksi dalam kegiatan perusahaan dapat terintegrasi.

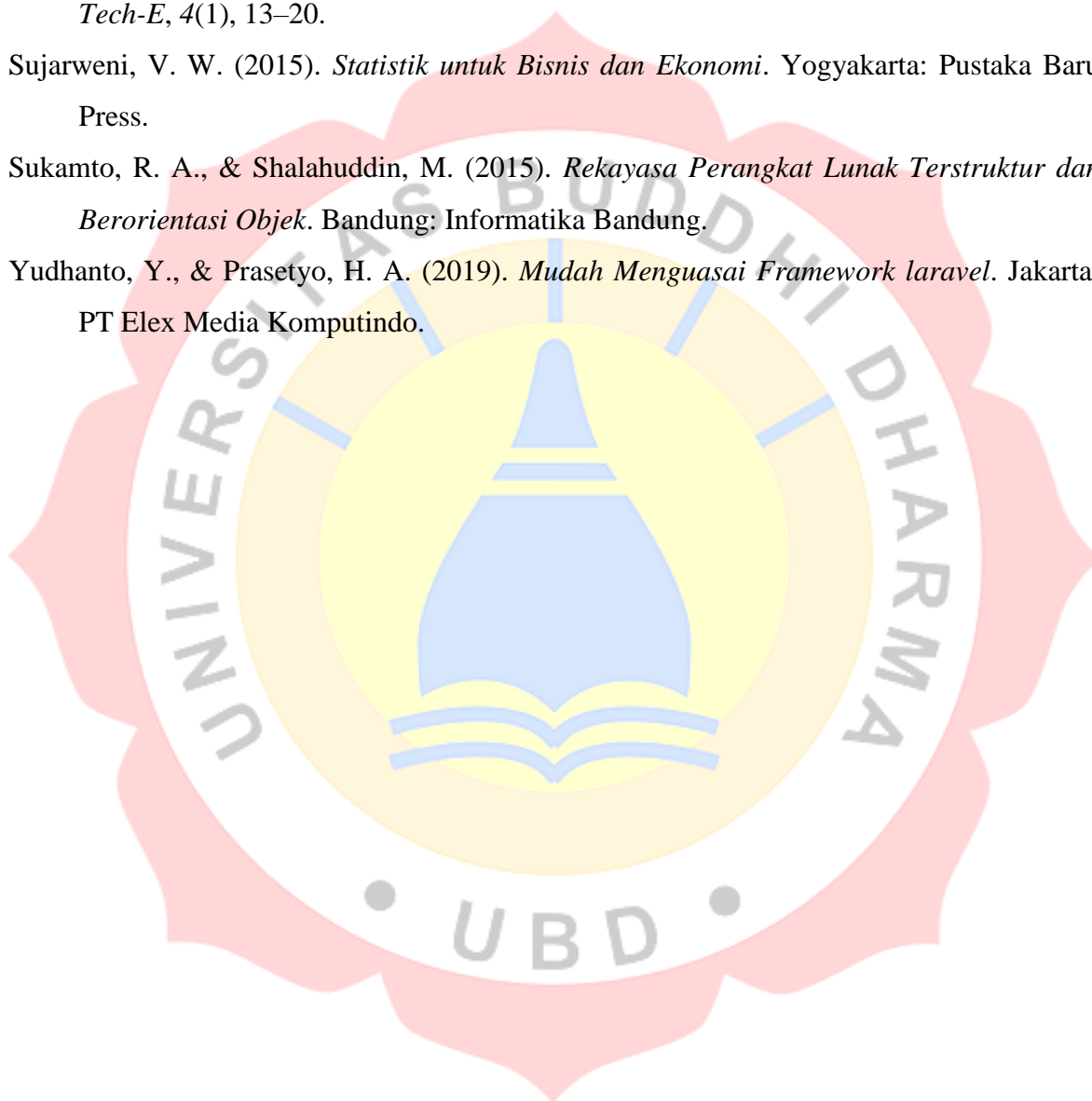
2. Perlu adanya pengawasan dan pengontrolan dalam pelaksanaan sistem ini sehingga dapat berjalan dengan baik dan sesuai dengan rencana, dan apabila terjadi kekurangan atau pun terjadi kesalahan pada sistem yang baru dapat segera dicari solusinya guna penyelesaian dan perbaikannya.
3. Perusahaan beralih dari sistem lama ke sistem yang baru yang lebih terintegrasi sehingga dapat memudahkan penjualan dan pemasaran perusahaan di masa yang akan datang.



DAFTAR PUSTAKA

- Anggraeni, E. Y., & Irviani, R. (2017). *Pengantar Sistem Informasi*. Yogyakarta: CV. Andi Offset.
- Dewi, F. S., P, H. P., & Susilowati, M. (2020). Perancangan Sistem Informasi Penjualan Online Distro Md Shoes Berbasis Website. *Jurnal Teknologi, Informasi Dan Industri*, 3(1),18–27.
- Dewi, P. D. S., & Santika, I. W. (2018). Pengaruh Technology Acceptance Model (TAM) Terhadap Niat Beli Ulang Online Di Kota Denpasar. *E-Jurnal Manajemen Unud*, 7(8), 4124–4152.
- Djahir, Y., & Pratita, D. (2015). *Bahan Ajar Sistem Informasi Manajemen*. Yogyakarta: Deepublish.
- Enterprise, J. (2015). *Pengenalan Visual Studio 2013*. Jakarta: PT Elex Media Komputindo.
- Fahmi, I. (2016). *Pengantar Manajemen Sumber Daya Manusia Konsep dan Kinerja*. Jakarta: Mitra Wacana Media.
- Haqi, B. (2019). *Aplikasi SPK Pemilihan Dosen Terbaik Metode Simple Additive Weighting SAW Dengan Java*. Yogyakarta: Deepublish.
- Haqi, B., & Setiawan, H. S. (2019). *Aplikasi Absensi Dosen dengan Java dan Smartphone sebagai Barcode Reader*. Jakarta: Elex Media Komputindo.
- Hutahaean, J. (2015). *Konsep Sistem Informasi*. Yogyakarta: Deepublish.
- Indonesia, E. (2022). Technology Acceptance Model. Retrieved from <https://educhannel.id/blog/artikel/technology-acceptance-model.html>
- Indrajani. (2015). *Database Design (Case Study All in One)*. Jakarta: PT Elex Media Komputindo.
- Junirianto, E. (2018). *Pemrograman Web Dengan Framework laravel*. Samarinda: Wade Group.
- Kurniawana, D. E., Saputrab, A., & Prasetyawanc, P. (2018). Perancangan Sistem Terintegrasi pada Aplikasi Siklus Akuntansi dengan Evaluasi Technology Acceptance Model. *Jurnal Resti*, 2(1), 315–321.
- Marsyadini, F., & Aprila, N. (2018). Niat Menggunakan E-commerce: Suatu Pendekatan Technology Acceptance Models (Studi Pada Usaha Mikro, Kecil dan Menengah Di Kota Bengkulu). *Jurnal Akutansi*, 8(1), 55–66.

- Mulyadi. (2016). *Sistem Informasi Akuntansi*. Jakarta: Salemba Empat.
- Munif, A. (2013). *Basis Data*. Jakarta: Kementerian Pendidikan & Kebudayaan.
- Nugroho, B. (2013). *Dasar Pemrograman Web PHP – MySQL dengan Dreamweaver*. Yogyakarta: Gava Media.
- Putra, A. Y., & Wijaya, A. H. (2020). Analysis and Design of E-Commerce (B2B) with Technology Acceptance Model Method at PT. Global Pharma Indonesia. *Journal Tech-E*, 4(1), 13–20.
- Sujarweni, V. W. (2015). *Statistik untuk Bisnis dan Ekonomi*. Yogyakarta: Pustaka Baru Press.
- Sukanto, R. A., & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Yudhanto, Y., & Prasetyo, H. A. (2019). *Mudah Menguasai Framework laravel*. Jakarta: PT Elex Media Komputindo.



DAFTAR RIWAYAT HIDUP



Data Pribadi

Nama Lengkap : Yoga Viriya
Tempat/Tanggal Lahir : Tangerang, 01 Juni 2000
Jenis Kelamin : Laki-laki
Alamat : Kp. Sawangan, Rt 03/03, Ds. Panongan, Kec. Panongan,
Kab. Tangerang - Banten
Agama : Buddha
Telepon : 0813-9200-0085
Email : yoga.viriya@gmail.com

Pendidikan Formal

2005 – 2011 : SDN 1 Panongan
2011 – 2014 : SMPN 1 Panongan
2014 – 2017 : SMK Atisa Dipamkara (Multimedia)
2018 – 2022 : Universitas Buddhi Dharma (Sistem Informasi)

Pengalaman Kerja

2017 – 2018 : PT. PBCM
2018 – Sekarang : PT. QIAO XING INDUSTRY

Tangerang, 22 Agustus 2022



Yoga Viriya

REQUIREMENT ELICITATION

10 responses + ⋮

Accepting responses

Summary Question Individual

Nama :

10 responses

- Marsel
- Riri
- Rian
- Septelya
- William
- Calvin
- Ria
- Talita
- Meirika

10 responses + ⋮

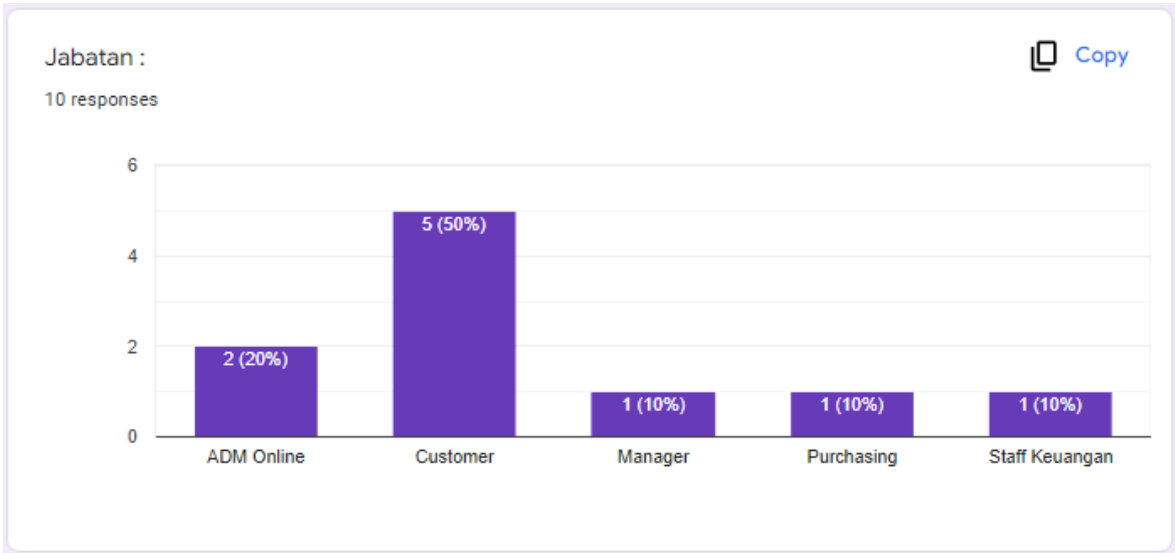
Accepting responses

Summary Question Individual

Nama :

10 responses

- Riri
- Rian
- Septelya
- William
- Calvin
- Ria
- Talita
- Meirika
- Lukman



- Saya Ingin Aplikasi Ini Dapat : Copy
- 10 responses
- Terdapat Halaman Daftar Produk
 - Terdapat Halaman Keranjang
 - Menampilkan Form Register
- Terdapat Halaman Dashboard
 - Bisa di Instal di Android
 - Terdapat Halaman Keranjang
 - Terdapat Halaman Produk
- Terdapat Halaman Produk
 - Ada di Playstore
 - Bisa di Instal di Android
- Terdapat Halaman Laporan Penjualan
 - Terdapat Halaman Dashboard
 - Terdapat Halaman Keranjang
- Menampilkan Form Register
 - Terdapat Halaman Laporan Penjualan

Saya Ingin Aplikasi Ini Dapat :

10 responses

Menampilkan Form Register
Terdapat Halaman Laporan Penjualan
Terdapat Halaman Dashboard

Terdapat Halaman Login
Terdapat Halaman Checkout Barang
Terdapat Halaman Keranjang

Menampilkan Form Register
Tampilan User Friendly

Terdapat Halaman Transaksi Penjualan
Terdapat Halaman Daftar Produk
Tampilan User Friendly

Terdapat Halaman Login
Terdapat Halaman Dashboard

Saya Ingin Aplikasi Ini Dapat :

10 responses

Terdapat Halaman Dashboard

Terdapat Halaman Login
Terdapat Halaman Checkout Barang
Terdapat Halaman Keranjang

Menampilkan Form Register
Tampilan User Friendly

Terdapat Halaman Transaksi Penjualan
Terdapat Halaman Daftar Produk
Tampilan User Friendly

Terdapat Halaman Login
Terdapat Halaman Dashboard

Terdapat Halaman Login
Tampilan User Friendly
Ada di Playstore

KUESIONER TAM (TECHNOLOGY ACCEPTANCE MODEL)

30 responses + ⋮

Accepting responses

Summary Question Individual

Nama :
30 responses

Meirika
Septelya
Arianto
Lee Janci
Santos
Talita
Calvin
Hari
John

30 responses + ⋮

Accepting responses

Summary Question Individual

Nama :
30 responses

Aziz
Irwan
Agus
Rahmat
Tomy
Ria
Amel
Gilang
Ibnu

30 responses



Accepting responses

Summary

Question

Individual

Nama :

30 responses

Rohman

Wahyu

William

Rian

Jordi

Marsel

Bayu

Lukman

Felix

30 responses



Accepting responses

Summary

Question

Individual

Nama :

30 responses

Rian

Jordi

Marsel

Bayu

Lukman

Felix

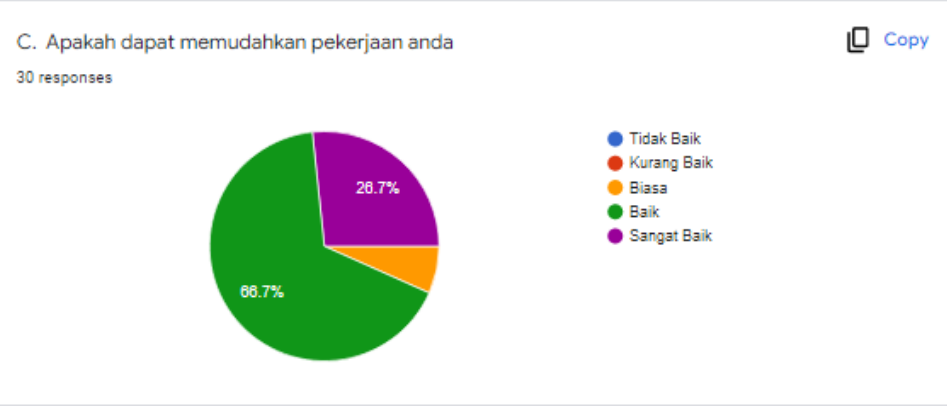
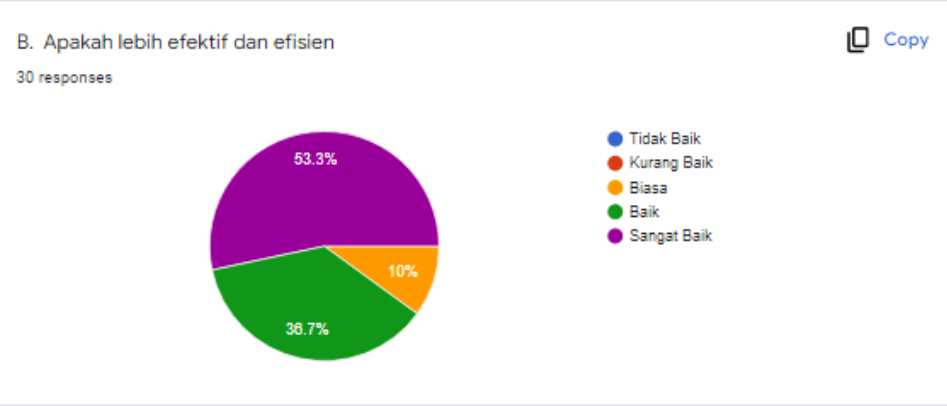
Ivan

Riri

Jaya



1. Perceived Usefulness

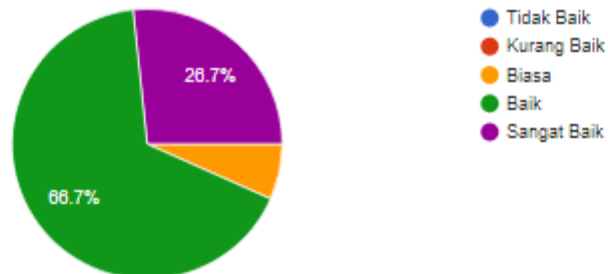


2. Perceived Ease of Use

A. Apakah mudah dimengerti dan dipahami

30 responses

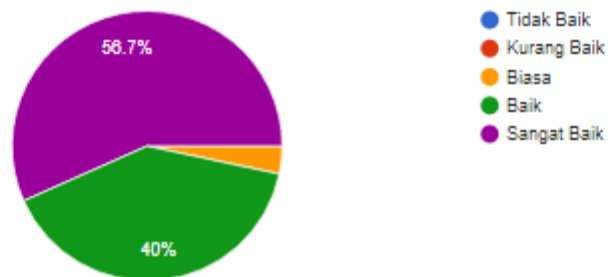
 Copy



B. Apakah aplikasi mudah digunakan

30 responses

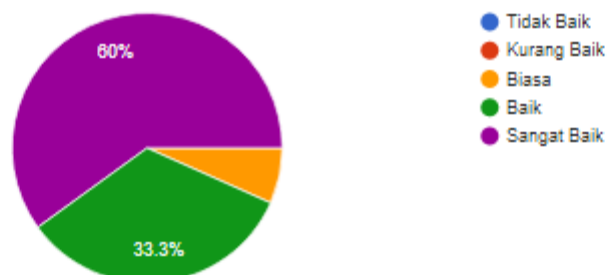
 Copy



C. Apakah anda cepat mahir dalam menggunakannya

30 responses

 Copy

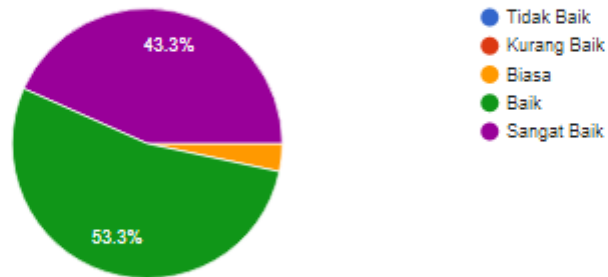


3. Usability

A. Mudah di ingat

30 responses

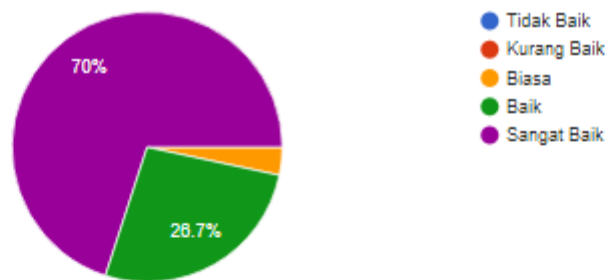
 Copy



B. Struktur menu mudah dipahami

30 responses

 Copy



C. Desain dan tata letak yang baik

30 responses

 Copy

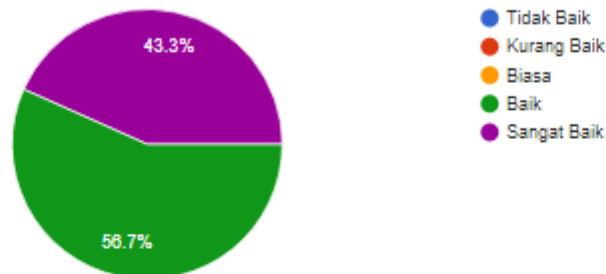


4. Graphic Design

A. Apakah text mudah di baca

 Copy

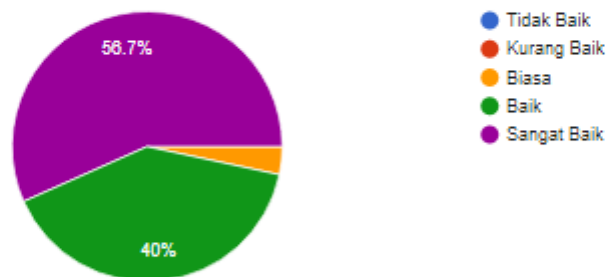
30 responses



B. Apakah komposisi warnanya baik dan konsisten

 Copy

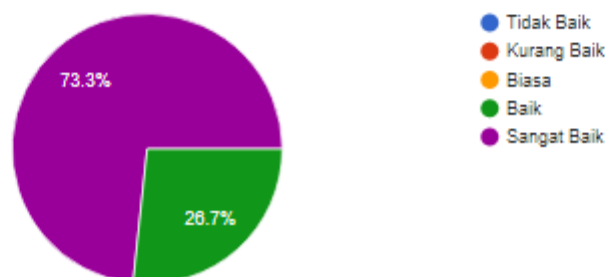
30 responses



C. Apakah penggunaan animasi sudah tepat


 Copy

30 responses

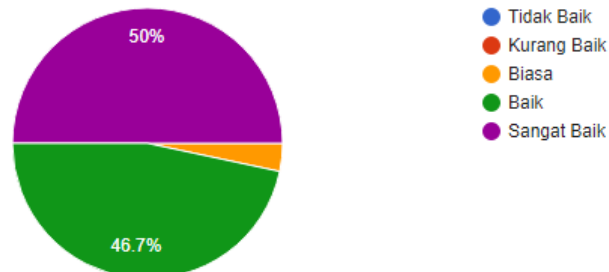


5. Compatibility

A. Dukungan browser (Mozilla Firefox, Google Chrome, Microsoft Edge)

 Copy

30 responses

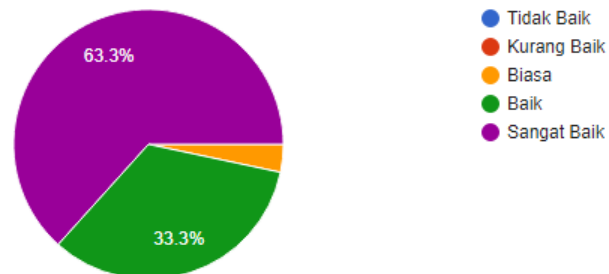


6. Functionality

A. Tombol button berfungsi dengan baik

 Copy

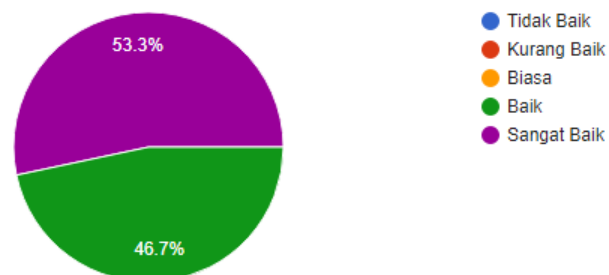
30 responses



B. Masing-masing menu berfungsi dengan baik

 Copy

30 responses





UNIVERSITAS BUDDHI DHARMA

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang

021 5517853 / 021 5586822 ✉ admin@buddhidharma.ac.id

KARTU BIMBINGAN TA/SKRIPSI

NIM : 20180700042
Nama Mahasiswa : YOGA VIRIYA
Fakultas : Sains dan Teknologi
Program Studi : Sistem Informasi
Jenjang : Strata Satu
Tahun Akademik/Semester : 2021/2022 Genap
Dosen Pembimbing : Tugiman, S.Kom., M.Kom
Judul Skripsi : ANALISIS DAN PERANCANGAN SISTEM INFORMASI E-COMMERCE PRODUK BERBASIS WEB DENGAN METODE TAM (TECHNOLOGY ACCEPTANCE MODEL)

Tanggal	Catatan	Paraf
2022-03-19	briefing mengenai skripsi dan pemilihan tema yang akan diambil pada penelitian	
2022-03-26	proses pengajuan judul dan deskripsi, proses upload judul di SIA dan proses persetujuannya, koreksi dari paper mahasiswa	
2022-04-02	proses pengajuan judul dan deskripsi, pembahasan bab 1	
2022-04-09	perbaiki judul, kerangka pemikiran, ringkasan jurnal, dan revisi	
2022-04-16	membahas mengenai jurnal pembanding, referensi, revisi bab 1 dan bab 2	
2022-06-05	Rvisi bab I, II, dan bahas bab III	
2022-06-25	bahas bab 4	
2022-06-27	activity diagram dan class diagram	
2022-06-29	bab 4 dan program	
2022-06-30	demo program	

Mengetahui
Ketua Program Studi

Benny Daniawan, M.Kom

Tangerang, 21 July 2022

Pembimbing

Tugiman, S.Kom., M.Kom



PT. QIAO XING INDUSTRY

Jl. Veteran Km. 1 No. 55 Desa Cukanggalih
Kecamatan Curug Kab. Tangerang – Banten 15810
Telp. (021) 59499698 Fax (021)59499697

SURAT KETERANGAN RISET

Yang bertanda tangan dibawah ini menerangkan bahwa:

Nama : Yoga Viriya
NIM : 20180700042
Mahasiswa dari : Universitas Buddhi Dharma
Program Studi : Sistem Informasi
Fakultas : Sains & Teknologi

Adalah benar telah melakukan penelitian di PT. QIAO XING INDUSTRY, sejak tanggal 15 April 2022 sampai dengan tanggal 01 Juli 2022.

Demikian surat keterangan ini agar dapat di penggunaan sebagaimana mestinya.

Tangerang, 01 Juli 2022

Hormat Kami

PT. QIAO XING INDUSTRY

Septelya Fairly
Manager



KEMENTERIAN KEUANGAN REPUBLIK INDONESIA
DIREKTORAT JENDERAL PAJAK

NPWP : 31.299.375.1-451.000

PT. QIAO XING INDUSTRY

JL. Veteran KM. 1, RT. 002/001 -
Cukang Galih-Curug Tangerang

TERDAFTAR : 25-03-2011
451

PERHATIAN

- Kartu ini harap disimpan baik-baik dan apabila hilang agar segera melapor ke Kantor Pelayanan Pajak terdaftar.
- NPWP agar dicantumkan dalam hal berhubungan dengan dokumen perpajakan.
- Dalam hal Wajib Pajak pindah domisili, supaya melaporkan diri ke Kantor Pelayanan Pajak lama atau Kantor Pelayanan Pajak Baru.

www.pajak.go.id



500200

BERSAMA ANDA MEMBANGUN BANGSA

LISTING PROGRAM

Admin.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Admin extends Model
{
    use HasFactory;
}
```

Auth.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Auth extends Model
{
    use HasFactory;
}
```

Category.php

```
<?php
namespace App\Models;
use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Category extends Model
{
    use HasFactory;
    protected $fillable = [
        'name',
        'url'
    ];
    protected $casts = [
        'created_at' => 'datetime:Y-m-d h:i:s',
    ];
}
```

```

        'updated_at' => 'datetime:Y-m-d h:i:s',
    ];
    public function getCreatedAtAttribute($value)
    {
        return Carbon::createFromTimestamp(strtotime($value))
            ->timezone('Asia/Jakarta')
            ->toDateTimeString();
    }
    public function getUpdatedAtAttribute($value)
    {
        return $this->getCreatedAtAttribute($value);
    }
    public function product()
    {
        return $this->belongsToMany(Product::class, 'product_categories');
    }
}

```

Customer.php

```

<?php
namespace App\Models;
use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
class Customer extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'user_id'
    ];
    public function user()
    {
        return $this->belongsTo(User::class);
    }
    public function getCreatedAtAttribute($value)
    {
        return Carbon::createFromTimestamp(strtotime($value))
            ->timezone('Asia/Jakarta')

```

```

        ->dateTimeString();
    }
    public function getUpdatedAtAttribute($value)
    {
        return $this->getCreatedAtAttribute($value);
    }
}

```

CustomerAddress.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class CustomerAddress extends Model
{
    use HasFactory;
}

```

Order.php

```

<?php
namespace App\Models;
use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\SoftDeletes;
class Order extends Model
{
    use HasFactory;
    use SoftDeletes;
    protected $fillable = [
        'customer_id', 'total',
        'return',
        'discount'
    ];
    protected $casts = [
        'created_at' => 'datetime:Y-m-d h:i:s',
        'updated_at' => 'datetime:Y-m-d h:i:s',
    ];
    public function getCreatedAtAttribute($value)

```

```

    {
        return Carbon::createFromTimestamp(strtotime($value))
            ->timezone('Asia/Jakarta')
            ->toDateTimeString();
    }
    public function getUpdatedAtAttribute($value)
    {
        return $this->getCreatedAtAttribute($value);
    }
    public function customer()
    {
        return $this->belongsTo(Customer::class);
    }
    public function orderDetail()
    {
        return $this->hasMany(OrderDetail::class);
    }
    public function orderPayment()
    {
        return $this->hasOne(OrderPayment::class);
    }
}

```

OrderDetail.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class OrderDetail extends Model
{
    use HasFactory;
    protected $fillable = [
        'order_id',
        'name',
        'price',
        'quantity',
        'imageUrl',
        'discount',
    ];
}

```

OrderPayment.php

```
<?php
namespace App\Models;
use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
class OrderPayment extends Model
{
    use HasFactory;
    protected $fillable = [
        'order_id',
        'amount',
        'status',
        'photo'
    ];
    public function getCreatedAtAttribute($value)
    {
        return Carbon::createFromTimestamp(strtotime($value))
            ->timezone('Asia/Jakarta')
            ->toDateTimeString();
    }
    public function getUpdatedAtAttribute($value)
    {
        return $this->getCreatedAtAttribute($value);
    }
}
```

PasswordHistory.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class PasswordHistory extends Model
{
    use HasFactory;
}
```

Permission.php

```
<?php
namespace App\Models;
use Laratrust\Models\LaratrustPermission;
class Permission extends LaratrustPermission
{
    public $guarded = [];
}
```

Product.php

```
<?php
namespace App\Models;
use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use Illuminate\Database\Eloquent\Factories\HasFactory;
class Product extends Model
{
    use HasFactory;
    use SoftDeletes;
    protected $fillable = [
        'name',
        'price',
        'unit',
        'photo'
    ];
    protected $casts = [
        'created_at' => 'datetime:Y-m-d h:i:s',
        'updated_at' => 'datetime:Y-m-d h:i:s',
    ];
    public function getCreatedAtAttribute($value)
    {
        return Carbon::createFromTimestamp(strtotime($value))
            ->timezone('Asia/Jakarta')
            ->toDateTimeString();
    }
    public function getUpdatedAtAttribute($value)
    {
        return $this->getCreatedAtAttribute($value);
    }
}
```


}

```
public function category()
{
    return $this->belongsToMany(Category::class, 'product_categories');
}
public function productImage()
{
    return $this->hasMany(ProductImage::class);
}
public function stock()
{
    return $this->hasOne(Stock::class);
}
}
```

ProductCategory.php

```
<?php

namespace App\Models;

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

class ProductCategory extends Model
{
    use HasFactory;
}
```

ProductPricingHistory.php

```
<?php

namespace App\Models;

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

class ProductPricingHistory extends Model
{
    use HasFactory;
}
```

}

Role.php

```
<?php
namespace App\Models;
use Laratrust\Models\LaratrustRole;
class Role extends LaratrustRole
{
    public $guarded = [];
}
```

Shipping.php

```
<?php

namespace App\Models;

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

class Shipping extends Model
{
    use HasFactory;
}
```

Stock.php

```
<?php

namespace App\Models;

use Carbon\Carbon;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Stock extends Model
{
    use HasFactory;

    protected $fillable = [
        'quantity',
    ]
}
```

```

        'product_id'
    ];
    protected $casts = [
        'created_at' => 'datetime:Y-m-d h:i:s',
        'updated_at' => 'datetime:Y-m-d h:i:s',
    ];
    public function getCreatedAtAttribute($value)
    {
        return Carbon::createFromTimestamp(strtotime($value))
            ->timezone('Asia/Jakarta')
            ->toDateTimeString();
    }
    public function getUpdatedAtAttribute($value)
    {
        return $this->getCreatedAtAttribute($value);
    }
    public function product()
    {
        return $this->belongsTo(Product::class);
    }
}

```

User.php

```

<?php
namespace App\Models;
use Carbon\Carbon;
use Laravel\Sanctum\HasApiTokens; use
Illuminate\Notifications\Notifiable;use
Laratrust\Traits\LaratrustUserTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use LaratrustUserTrait;
    use HasApiTokens; use
    HasFactory;
    use Notifiable;

```

```

/**
 * The attributes that are mass assignable.
 *
 * @var string[]
 */
protected $fillable = [
    'name',
    'email',
    'is_admin',
    'password',
];

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

/**
 * The attributes that should be cast.
 *
 * @var array
 */
protected $casts = [
    'email_verified_at' => 'datetime',
    'created_at' => 'datetime:Y-m-d h:i:s',
    'updated_at' => 'datetime:Y-m-d h:i:s',
];

public function getCreatedAtAttribute($value)
{
    return Carbon::createFromTimestamp(strtotime($value))
        ->timezone('Asia/Jakarta')
        ->toDateTimeString();
}

public function getUpdatedAtAttribute($value)
{

```

```

        return $this->getCreatedAtAttribute($value);
    }
    public function customer()
    {
        return $this->hasOne(Customer::class);
    }
}

```

CategoryController.php

```
<?php
```

```

namespace App\Http\Controllers;

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

class CategoryController 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 = Category::orderBy('id', 'desc')->paginate(10);
    $this->header = 'List Category';
    $this->currentNav = 'Product & Stock';
    return Inertia::render('Report', $this->returnProps([
        'table' => [
            'title' => 'List Category',
            'pageRoute' => 'category.index',
            'action' => 'category',
            'list' => $data,
            'header' => [
                [
                    'Header' => '#Id',
                    'accessor' => 'id',
                ],[
                    'Header' => 'Category Name',
                    'accessor' => 'name',
                ],[
                    'Header' => 'Photo Category',
                    'accessor' => 'url',
                    '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 store(Request $request)
{
    $this->validate($request, [
        'name' => 'required|unique:products,name',
        'image' => 'required|image|mimes:png,jpg',
    ]);
    $productPhoto = $request->file('image')->getClientOriginalName();
    $photoName = date('dmy') . $request->name . uniqid() . $productPhoto;
    Category::create([
        'name' => $request->name, 'url'
        => $photoName,
    ]);
    $request->image->move(public_path('storage/images'), $photoName);
    return redirect()->route('category.index');
}
public function create()
{
    $this->currentNav = 'Product & Stock';
    $this->header = 'List Category';
    return Inertia::render('Form', $this->returnProps([
        'form' => [
            'title' => 'Create new Category',
            'action' => route('category.store'),
            'design' => [
                [

```

```

        'type' => 'text',
        'label' => 'Category Name',
        'name' => 'name',
        'require' => true,
    ],[
        'type' => 'photo',
        'label' => 'Category Image',
        'name' => 'image',
        'require' => true,
    ]
    ]
    ]);
}
public function edit(Category $category)
{
    $this->currentNav = 'Product & Stock';
    $this->header = 'List Category';
    return Inertia::render('Form', $this->returnProps([
        'form' => [
            'title' => 'Create new Category',
            'action' => route('category.update', $category),
            'type' => 'update',
            'design' => [[
                'type' => 'text',
                'label' => 'Category Name',
                'name' => 'name',
                'value' => $category->name,
                'require' => true,
            ],[
                'type' => 'photo',
                'label' => 'Category Image',
                'name' => 'image',
                'value' => $category->url,
                'require' => true,
            ]
        ]
    ])
}

```

```

        ]
    ));
}
public function update(Request $request, Category $category)
{
    $this->validate($request, [
        'name' => 'required|unique:products,name',
        'image' => 'image|mimes:png,jpg',
    ]);
    $category->name = $request->name;if
    ($request->image) {
        $productPhoto = $request->file('image')->getClientOriginalName();
        $photoName = date('dmy') . $request->name . uniqid() . $productPhoto;
        $category->url = $photoName;
        $request->image->move(public_path('storage/images'), $photoName);
    }
    $category->save();
    return redirect()->route('category.index');
}
public function destroy(Category $category)
{
    $category->delete();
    return Redirect::route('category.index');
}
}

```