**PPL IF14A**

**OPERKREDIT.ID**

**GenerICT :**

1. **Muhammad Iqbal Pratama 0102514021**
2. **Umiyati 0102514021**
3. **Endah Sulisthyani 0102516902**

**FAKULTAS SAINS DAN TEKNOLOGI**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS AL AZHAR INDONESIA**

**2017**

|  |
| --- |
| **Versi 3.0** |
| 1. **Project Plan** |
| 1. **System Proposal** |
| 1. **Design Report** |

# PUBLICATION DETAIL

**Rekam perubahan**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**DESIGN REPORT**

# DAFTAR ISI

[PUBLICATION DETAIL ii](#_Toc496697833)

[DAFTAR ISI iv](#_Toc496697834)

[DAFTAR GAMBAR v](#_Toc496697835)

[DAFTAR TABEL vi](#_Toc496697836)

[DAFTAR LAMPIRAN vii](#_Toc496697837)

[A. SYSTEM DESIGN AND DEVELOPMENT STRATEGY 1](#_Toc496697838)

[1. Strategi Integrasi 1](#_Toc496697839)

[2. Strategi Migrasi dan Konversi 1](#_Toc496697840)

[B. CLASS AND METHOD DESIGN 2](#_Toc496697841)

[1. Class Diagram 2](#_Toc496697842)

[C. Database Design 4](#_Toc496697843)

[1. Database Schema 4](#_Toc496697844)

[2. Table Structures and Attributes 4](#_Toc496697845)

[D. User Interface DESIGN 12](#_Toc496697846)

[1. Site Map 12](#_Toc496697847)

[2. Graphical User Interface Design 13](#_Toc496697848)

[E. PHYSICAL Architecture DESIGN 17](#_Toc496697849)

[1. Type of Architecture 17](#_Toc496697850)

[2. Hardware and Software Specification 17](#_Toc496697851)

[3. Alternative Teknologi 18](#_Toc496697852)

[F. Test Plan 24](#_Toc496697853)

[1. [TestName Plan] 24](#_Toc496697854)

[DAFTAR PUSTAKA 25](#_Toc496697855)

# DAFTAR GAMBAR

[Gambar 1 Class Diagram 2](#_Toc496698428)

[Gambar 2 Relationship Diagram 4](file:///D:\My%20Future\Kuliah\SEMESTER%207\ppl\Template%20System%20Specification%20(Design)_IF14A.docx#_Toc496698429)

[Gambar 3 Struktur Menu operkredit.id 12](#_Toc496698430)

[Gambar 4 Mockup General Home Page 13](file:///D:\My%20Future\Kuliah\SEMESTER%207\ppl\Template%20System%20Specification%20(Design)_IF14A.docx#_Toc496698431)

[Gambar 5 Mockup Registrasi Pengunjung 13](#_Toc496698432)

[Gambar 6 Mockup Form login 14](#_Toc496698433)

[Gambar 7 Mockup Profile 14](#_Toc496698434)

[Gambar 8 Mockup Registered Home Page 15](#_Toc496698435)

[Gambar 9 Mockup Detail Barang 15](#_Toc496698436)

[Gambar 10 Mockup Tambah Barang dan lihat daftar barang 16](#_Toc496698437)

[Gambar 11 Mockup form input barang 16](#_Toc496698438)

[Gambar 12 Physical Architecture **Error! Bookmark not defined.**](#_Toc496698439)

# DAFTAR TABEL

[Tabel 1 Framework 18](#_Toc496700512)

[Tabel 2 Programming Language 18](#_Toc496700513)

[Tabel 3 DBMS 20](#_Toc496700514)

# DAFTAR LAMPIRAN

**No table of figures entries found.**

# SYSTEM DESIGN AND DEVELOPMENT STRATEGY

## Strategi Integrasi

Rencana integrasi yang kami pilih yaitu :

## Strategi Migrasi dan Konversi

Selama ini penawaran oper kredit hanya dilakukan secara konvensional, dan masyarakat pun belum semua mengerti mekanismenya. Di sisi lain, sistem kredit semakin lama semakin banyak digunakan oleh masyarakat. Produknya pun sangat variatif, dari mulai rumah, mobil/motor, hingga handphone dan barang-barang konsumsi lainnya. Khusus untuk produk yang bernilai asset (akan selalu mengalami kenaikan harga), kesempatan ini pun bisa jadi sarana baru yang sangat baik untuk berinvetasi. Oleh karena itu, kami membuat system migrasi dari penawaran oper kredit yang dilakukan secara konvensional ke system konversi yang lebih menarik. Sehingga kami membuat web operkredit.id dengan memudahkan customer untuk melakukan mengoper barang dan mengkredit barang

# CLASS AND METHOD DESIGN

## Class Diagram

Rancangan *Class Diagram* untuk fase desain adalah sebagai berikut :



Gambar 1 *Class Diagram*

Penjelasan Class Diagram:

* Pengunjung : merupakan *parent* dari kelas Pengambi\_kredit dan Pengoper\_kredit yang memiliki beberapa atribut (username, nama, email, jenis\_kelamin, no\_hp) dan beberapa operasi (lihat\_barang(), update\_pengunjung(), lihat\_history(), lihat\_jadwal(), verifikasi\_akun()). Atribut dan operasi tersedut diwariskan kepada anaknya.
* Pengambil\_kredit: merupakan *child* dari Pengunjung yang mewarisi atribut-atribut dan operasi-operasi yang dimiliki oleh *parent-*nya serta memiliki atribut khusus yang tidak dimiliki oleh *child* lain yaitu id\_pengambil\_kredit dan beberapa operasi yaitu input\_pengambilan\_produk(),update\_pengambilan\_produk(),pilih\_metode\_pembayaran(), mencari\_produk(), menyetujui\_over\_kredit()
* Pengover\_kredit : merupakan *child* dari Pengunjung yang mewarisi atribut-atribut dan operasi-operasi yang dimiliki oleh *parent-*nya serta memiliki atribut khusus yang tidak dimiliki oleh *child* lain yaitu id\_pengoper\_kredit, alamat, kota, dan provinsi serta memiliki beberapa operasi yaitu input\_barang(), update\_barang(), dan konfirmasi\_transaksi().
* Barang : merupakan kelas yang memiliki atribut id\_barang, nama\_barang warna, berat, ukuran, deskripsi, harga\_dp\_kembali, harga\_cicilan dan sisa\_cicilan. Kelas ini memiliki operasi yaitu kategori() yang mengambil nilai dari kelas Kategori dan barang() yang mengeluarkan atribut barang.
* Kategori : kelas ini menjelaskan kategori yang dimiliki oleh masing-masing barang dengan atribut id\_kategori dan nama\_kategori serta memiliki operasi kategori() yang mengeluarkan nilai dari kategori.
* Tim\_pengesahan : kelas ini memberikan keterangan untuk setiap transaksi yang dilakukan atas persetujuan dari tim\_pengesahan. Kelas ini memiliki atribut id\_tim\_pengesahan dan nama\_tim\_pengesahan. Operasi yang dilakukan oleh kelas ini meliputi lihat\_transaksi\_pengambilan\_kredit(), input\_jadwal(), dan menyetujui\_pemindahan\_kredit()
* Pengesahan : kelas ini menberisi beberapa atribut yaitu no\_pengesahan dan file\_pengesahan(). Kelas ini mengambil nilai dari Tim\_pengesahan untuk menjelaskan pengesahan dilakukan oleh siapa.
* Pengambilan\_kredit : kelas ini memiliki beberapa atribut yaitu id\_pengambilan\_kredit dan tanggal serta memiliki operasi set\_detail\_pengambilan\_kredit().
* Detail\_pengambilan\_kredit : kelas ini memberikan penjelaskan terhadap kelas Pengambilan\_kredit dengan melist detail dari barang yang diambil. Kelas ini memiliki atribut yaitu id\_detail\_pengambilan\_kredit, jumlah, harga, dan total\_harga. Dan memiliki beberapa operasi yaitu barang() dan return\_total\_harga().
* Metode\_pembayaran : merupakan kelas yang berisi dari metode-metode pembayaran yang dipilih oleh pengambil kredit saat melakukan pembayaran. Kelas ini memiliki atribut id\_metode\_pembayaran dan nama\_metode\_pembayaran.

# Database Design

Berikut merupakan penjelasan mengenai skema database dari operkredit.id yang telah dibuat:

## Database Schema

Gambar 2 *Relationship Diagram*

## Table Structures and Attributes

Berikut merupakan keterangan struktur data dari hasil basis data yang telah dibuat.

###### Struktuk data pengunjung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : pengunjung | | | | |
| isi : data pengunjung | | | | |
| primary key : username | | | | |
| foreign key : - | | | | |
| **No** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | username\* | varchar | 12 | username pengunjung |
| 2 | nama\_depan | varchar | 30 | nama depan pengunjung |
| 3 | nama\_belakang | varchar | 30 | nama belakang pengunjung |
| 4 | jenis\_kelamin | varchar | 1 | jenis kelamin pengunjung |
| 5 | no\_hp | varchar | 12 | kontak pengunjung |

###### Struktur data Pengoper\_kredit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : pengoper\_kredit | | | | |
| isi : data pengoper\_kredit | | | | |
| primary key : id\_pengoper\_kredit | | | | |
| foreign key : username | | | | |
| **No** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_pengoper\_kredit\* | varchar | 12 | kode untuk pengoper kredit |
| 2 | username\*\* | varchar | 12 | *foreign key* dari tabel pengunjung |
| 3 | alamat | varchar | 30 | alamat pengooper kredit sekaligus barang |
| 4 | kota | varchar | 30 | kota pengoper kredit |
| 5 | provinsi | varchar | 30 | provinsi pengoper kredit |

###### Struktur data Pengambil\_kredit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : pengambil\_kredit | | | | |
| isi : data pengambil kredit | | | | |
| primary key : id\_pengambil\_kredit | | | | |
| foreign key : username | | | | |
| **No** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_pengambil\_kredit\* | varchar | 12 | kode untuk pengambil kredit |
| 2 | username\*\* | varchar | 12 | *foreign key* dari tabel pengunjung |

###### Struktur data Pengambilan\_kredit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : pengambilan\_kredit | | | | |
| isi : data pengambilan kredit | | | | |
| primary key : id\_pengambilan\_kredit | | | | |
| foreign key : id\_pengambil\_kredit, id\_metode\_pembayaran, no\_pengesahan | | | | |
| **No** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_pengambilan\_kredit\* | varchar | 12 | kode untuk setiap pengambilan kredit |
| 2 | id\_pengambil\_kredit\*\* | varchar | 12 | *foreign key* dari tabel pengambil kredit |
| 3 | id\_metode\_pembayaran\*\* | varchar | 12 | *foreign key* dari tabel metode pembayaran |
| 4 | no\_pengesahan\*\* | varchar | 12 | *foreign key* dari tabel pengesahan |
| 5 | tanggal | date |  | tanggal pengambilan kredit |

###### Struktur data Detail\_pengambilan\_kredit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : detail\_pengambilan\_kredit | | | | |
| isi : data detail pengambilan kredit | | | | |
| primary key : id\_detail\_pengambilan\_kredit | | | | |
| foreign key : id\_pengambilan\_kredit, id\_barang | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_detail\_pengambilan\_kredit\* | varchar | 12 | kode untuk setiap item pengambilan kredit |
| 2 | id\_pengambilan\_kredit\*\* | varchar | 12 | *foreign key* dari tabel pengambilan kredit |
| 3 | id\_barang\*\* | varchar | 12 | *foreign key* dari tabel barang |
| 4 | jumlah | int |  | jumlah untuk setiap barang |
| 5 | harga | int |  | harga barang |
| 6 | total\_harga | int |  | total harga |

###### Struktur data Barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : barang | | | | |
| isi : data barang yang dioper | | | | |
| primary key : id\_barang | | | | |
| foreign key : id\_kategori | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_barang\* | varchar | 12 | kode untuk setiap barang |
| 2 | id\_kategori \*\* | varchar | 12 | *foreign key* dari tabel kategori |
| 3 | nama\_barang | varchar | 30 | nama barang |
| 4 | ukuran | int |  | ukuran dari barang |
| 5 | warna | varchar | 15 | warna barang |
| 6 | berat | varchar | 10 | berat barang |
| 7 | deskripsi | text |  | deskripsi tambahan tentang barang |
| 8 | harga\_dp\_kembali | int |  | nominal yang harus dibayar oleh pengambil kredit kepada pengoper kredit sebagai pengganti dp |
| 9 | harga\_cicilan | int |  | besar cicilan yang harus dibayar |
| 10 | sisa\_cicilan | varchar | 20 | jumlah periode waktu sisa cicilan |
| 11 | id\_pengoper\_kredit | varchar | 12 | *foreign key* dari tabel pengoper\_kredit yang menjelsakan kepemilikan barang |

###### Struktur data Kategori

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : kategori | | | | |
| isi : data kategori barang | | | | |
| primary key : id\_kategori | | | | |
| foreign key : - | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_kategori\* | varchar | 12 | kode untuk setiap kategori barang |
| 2 | nama\_kategori | varchar | 30 | nama kategori |

###### Struktur data Metode\_pembayaran

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : metode pembayaran | | | | |
| isi : data metode pembayaran yang disediakan | | | | |
| primary key : id\_metode\_pembayaran | | | | |
| foreign key : - | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_metode\_pembayaran\* | varchar | 12 | kode untuk setiap metode pembayaran |
| 2 | nama\_metode\_pembayaran | varchar | 30 | nama metode pembayaran |

###### Struktur data Pengesahan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : pengesahan | | | | |
| isi : data pengesahan | | | | |
| primary key : no\_pengesahan | | | | |
| foreign key : id\_tim\_pengesahan | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | no\_pengesahan \* | varchar | 12 | kode untuk setiap file |
| 2 | id\_tim\_pengesahan\*\* | varchar | 12 | *foreign key* dari tabel tim pengesahan |
| 3 | file\_pengesahan | varchar | 30 | dokumen serah terima antara pengambil dan pengoper kredit |

###### Struktur data Tim\_pengesahan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : tim pengesahan | | | | |
| isi : data tim pengesahan | | | | |
| primary key : id\_tim\_pengesahan | | | | |
| foreign key : - | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | id\_tim\_pengesahan \* | varchar | 12 | kode untuk setiap tim pengesahan |
| 2 | nama\_tim\_pengesahan | varchar | 30 | nama tim pengesahan |

###### Struktur data jadwal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nama file : jadwal | | | | |
| isi : data jadwal pertemuan antara pengambil dan pengoper kredit | | | | |
| primary key : id\_jadwal | | | | |
| foreign key : id\_jadwal, id\_pengambilan\_kredit | | | | |
| **no** | **field name** | **type** | **field size** | **Keterangan** |
| 1 | Id\_jadwal \* | varchar | 12 | kode untuk setiap jadwal |
| 2 | id\_pengambilan\_kredit \*\* | varchar | 12 | *foreign key* dari tabel pengambilan­\_kredit |
| 3 | tanggal | date |  | Tanggal bertemunya pengoper dan pengambil kredit |
| 4 | waktu | time |  | waktu bertemunya pengoper dan pengambil kredit |

# User Interface DESIGN

Berikut merupakan penjelasan mengenai tampilan antar muka yang dikembangkan

## Site Map

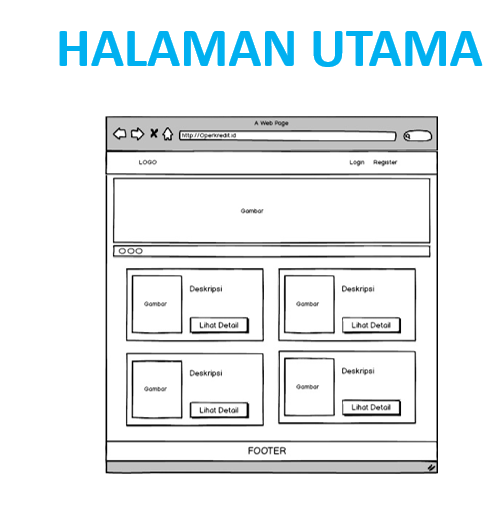
pemetaan halaman sistem yang dapat diakses adalah sebagai berikut:



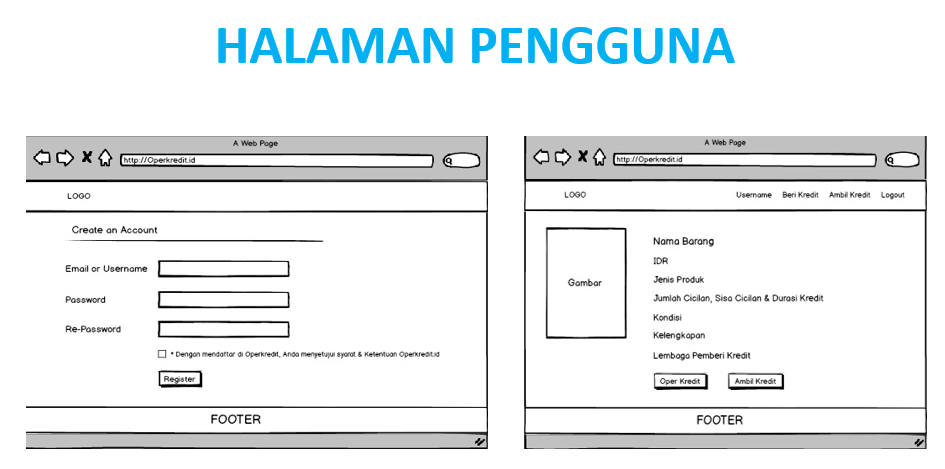
Gambar 3 Struktur Menu operkredit.id

## Graphical User Interface Design

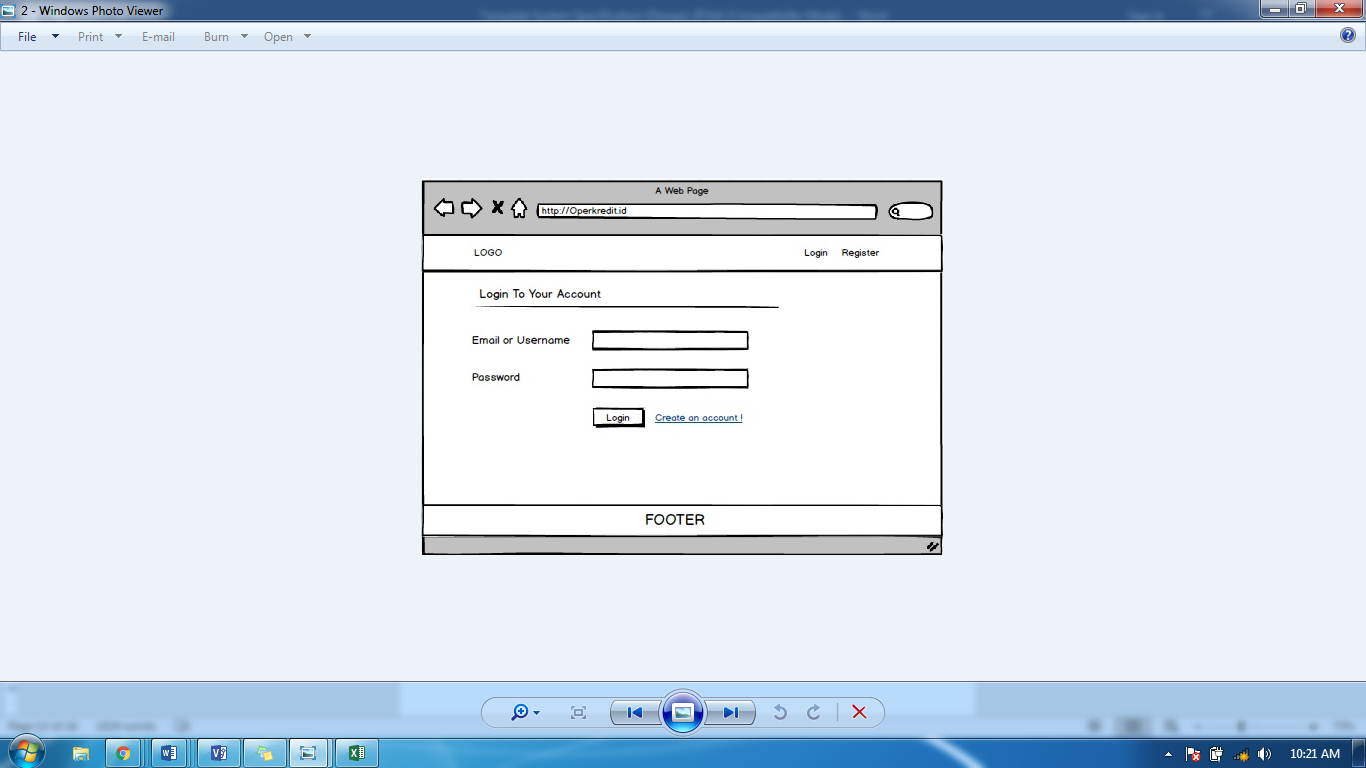
Berikut merupakan *mockup GUI* yang telah didesain untuk operkredit.id:



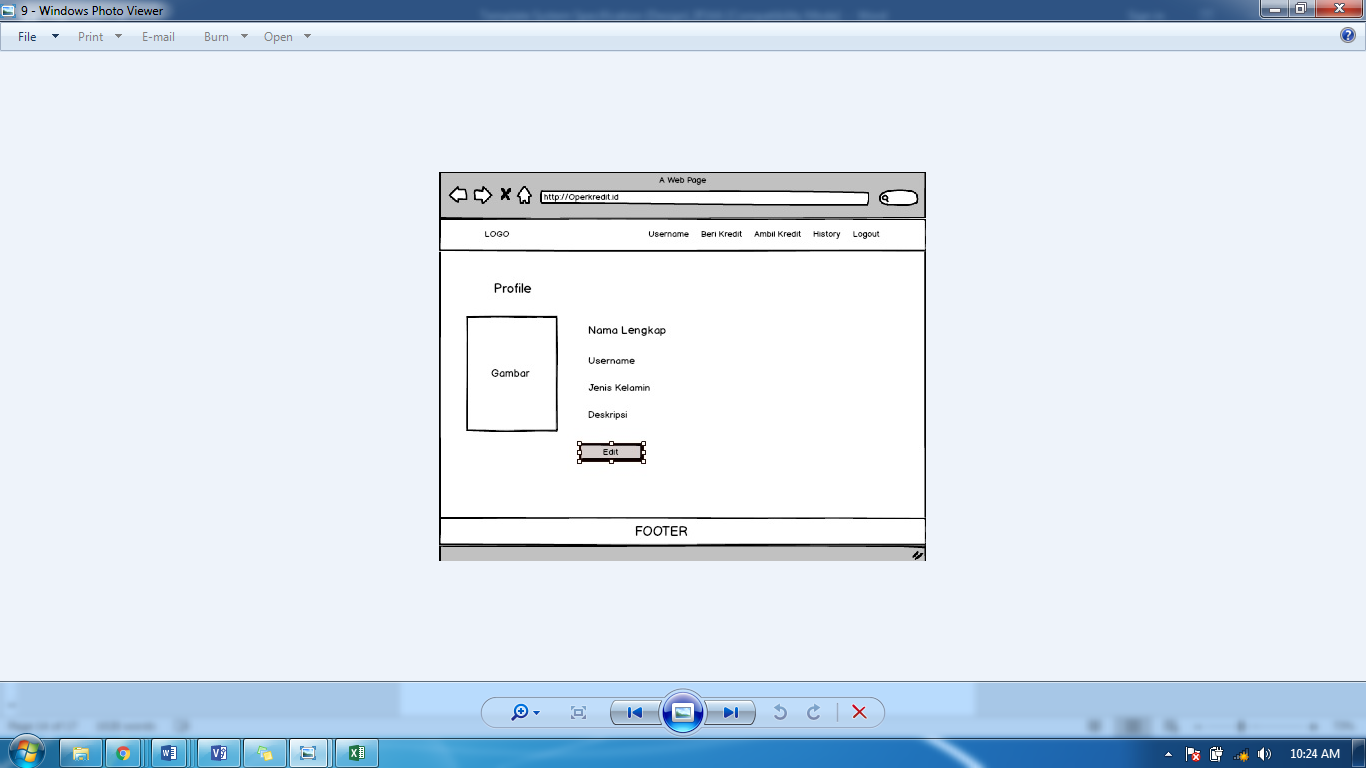
Gambar 4 *Mockup* *General Home Page*



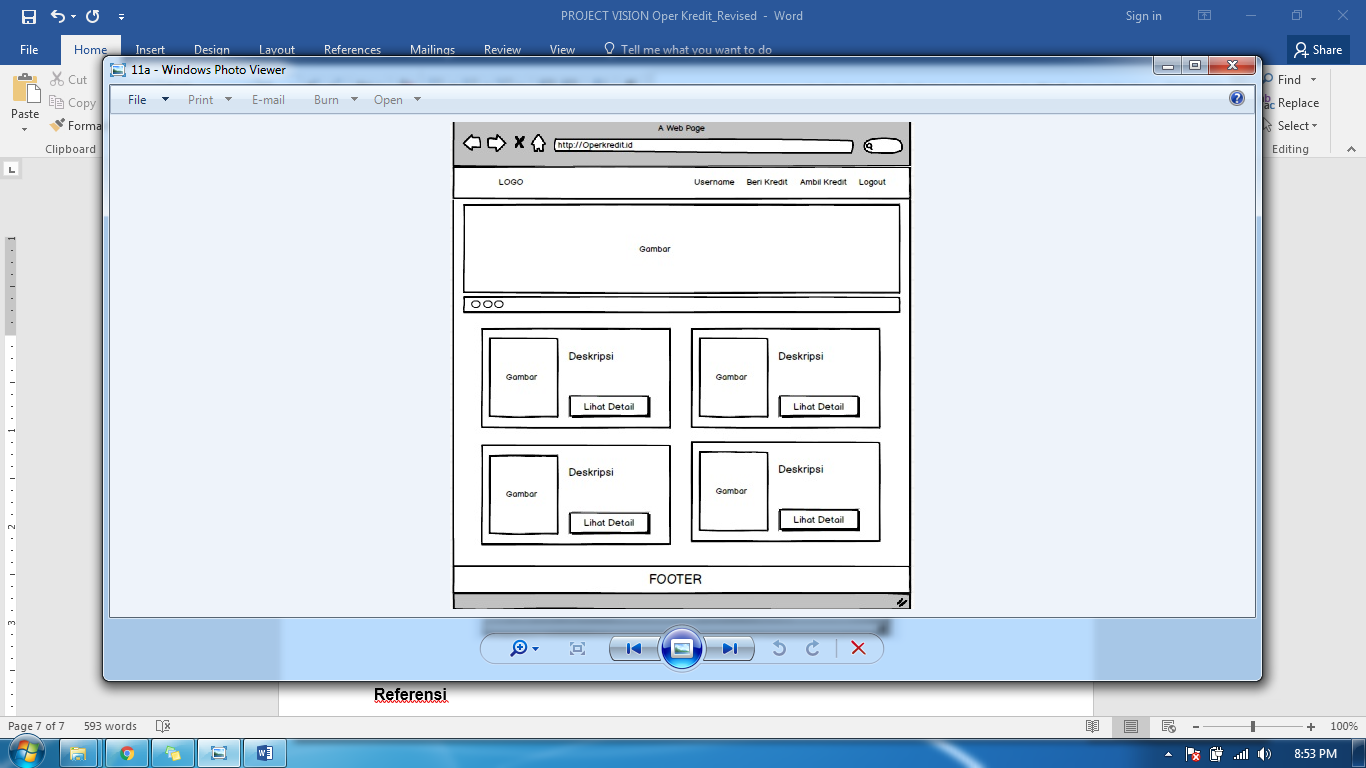
Gambar 5 *Mockup* Registrasi Pengunjung



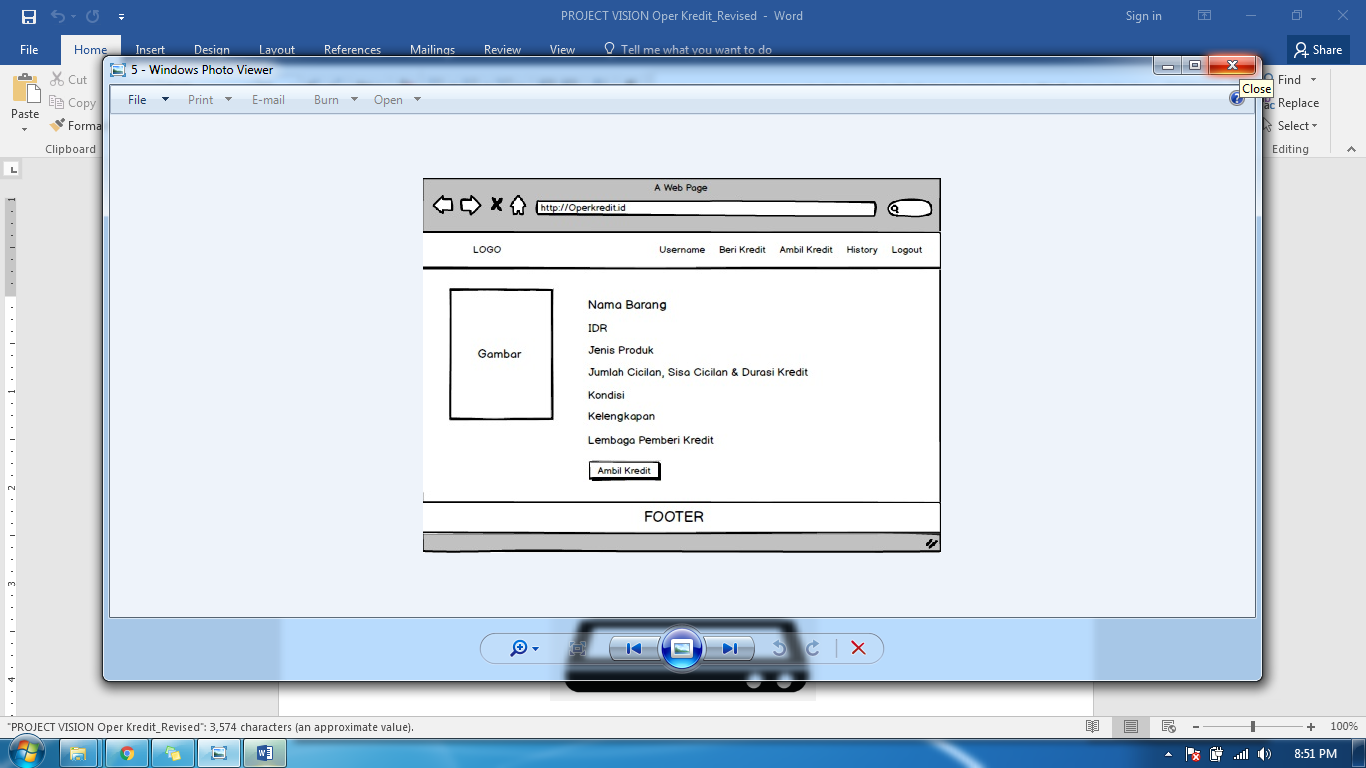
Gambar 6 *Mockup* *Form login*



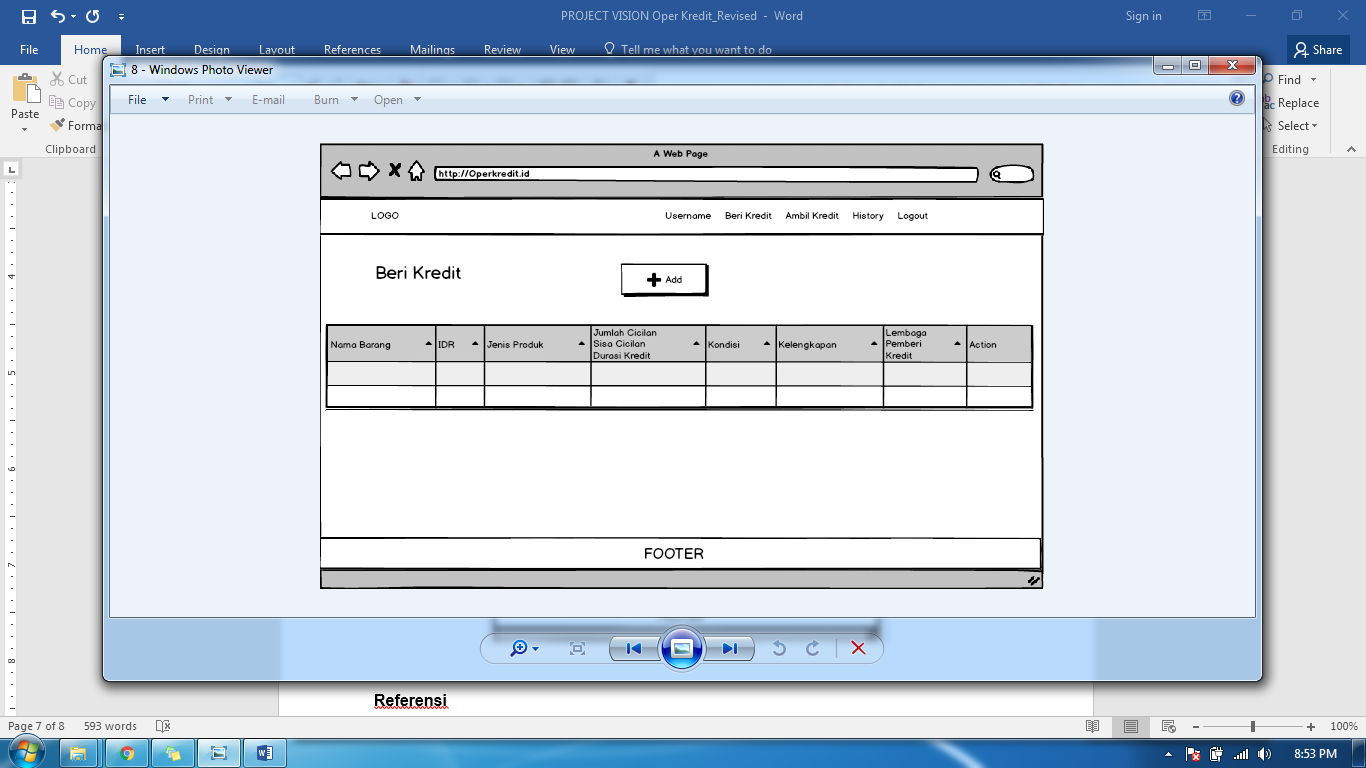
Gambar 7 *Mockup Profile*



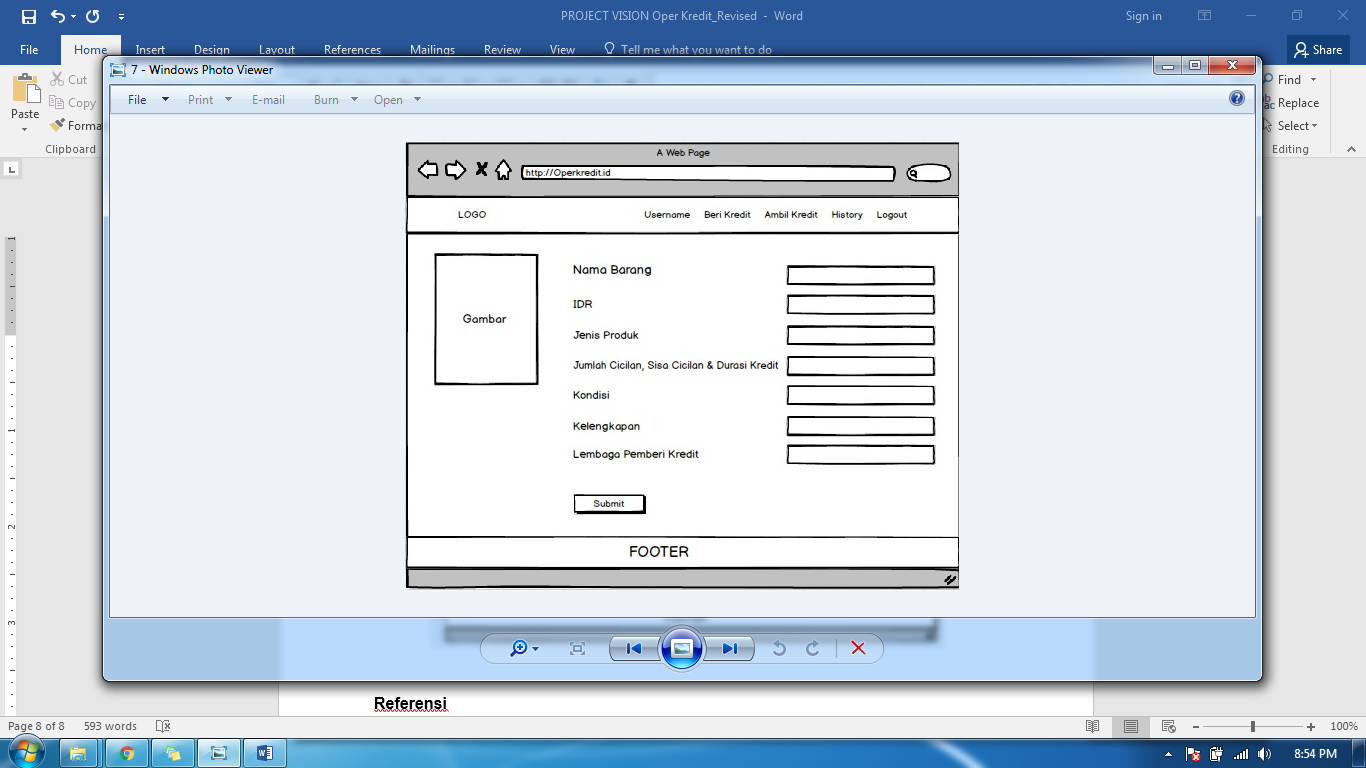
Gambar 8 *Mockup Registered Home Page*



Gambar 9 *Mockup Detail Barang*



Gambar 10 *Mockup* Tambah Barang dan lihat daftar barang

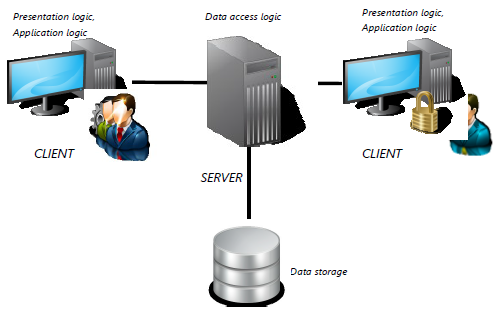


Gambar 11 *Mockup form input barang*

# PHYSICAL Architecture DESIGN

## Type of Architecture

*Operkredit.Id menggunakan desain three tier client-server architecture, yang terdiri dari presentation logic, application logic, dan data access logic. Untuk presentation logic dan application logic akan dieksekusi pada komputer client, sedangkan data acces logic, dan data storage dilakukan pada server. Dengan kata lain,* Three client-server *berkonsep* client *hanya menampilkan* presentation logic*,* application logic *hanya menangani* business logic *dan* database server *menangani* data access logic *dan* data storage.



(sumber gambar: dokumen Salihara)

Gambar Physical Architecture

## Hardware and Software Specification

Bagian ini menjelaskan spesifikasi hardware dan software yang dibutuhkan untuk pengembangan, deployment, dan pemeliharaan sistem.

###### Hardware Specification

Spesifikasi hardware yang dibutuhkan untuk pengembangan, deployment dan pemeliharaan system kami yaitu :

**Laptop**

Windows Edition : Windows 8.1 Pro

Processor : Intel® Core ™ i3-4030U CPU @ 1.90Hz 1.90Hz

Installed memory (RAM) : 4.00 GB

System type : 64-bit Operating System, x64-based processor

###### Software Specification

Software yang dibutuhkan untuk membangun modul terkait yaitu :

-Operating System : Windows Server 2012

-Laravel

-MySQL

-Notepad++ dan sublime text

-XAMPP

## Alternative Teknologi

###### Framework

Tabel *Framework*

|  |  |  |
| --- | --- | --- |
| Modul | Laravel | Codeigniter |
| Class Auto Loading | Yes | No |
| * Database mySQL | Yes | Yes |
| * Database SQLite | Yes | Yes |
| * Database MSSQL | Yes | Yes |
| * Database PostgreSQL | Yes | Yes |
| * Database Cubrid | Yes | No |
| * ODBC drivers | Yes | No |
| * Database MariaDB | No. | No |
| Authentication Library | Yes | No |
| External Modules | Yes (Bundles) | No |
| Form Validation Rules | Yes | Yes |
| Internationalization | Yes | Yes |
| Layout Control | Yes | No |
| ORM | Yes | No |
| Error Stack Trace | Yes | No |

###### Programing Language

Tabel *Programming Language*

|  |  |  |
| --- | --- | --- |
| PHP | Ruby | Java |
| PHP bersifat open source, berfungsi utaman digunakan untuk pengembangan web (a.k.a. membuat halaman web.) PHP diciptakan untuk merampingkan pembuatan halaman web. Ini adalah bahasa pemrograman yang cukup sederhana yang bisa dijemput dengan cepat. Ini digunakan oleh banyak perusahaan berbasis web termasuk Facebook, Wikipedia, dan WordPress. | Ruby adalah bahasa pemrograman berorientasi objek yang berorientasi pada tujuan yang dikembangkan pada pertengahan tahun 90an di Jepang. Ini adalah salah satu bahasa pemrograman yang lebih sederhana untuk dipelajari dan sering digunakan sebagai batu loncatan untuk Ruby on Rails. Ini dikembangkan untuk menjadi menyenangkan sekaligus kode, dan untuk meningkatkan produktivitas. Ini dikenal karena mudah dibaca, dan akibatnya banyak programmer merekomendasikan belajar Ruby sebagai bahasa pengkodean pertama Anda. Ini digunakan oleh situs seperti Hulu, Shopify, Airbnb dan banyak lainnya. | Java saat ini adalah bahasa yang paling populer dan banyak digunakan di dunia. Meskipun awalnya dibuat untuk TV interaktif, ini dikenal sebagai bahasa pilihan untuk perangkat Android. Ini juga bahasa pengkodean perangkat lunak tingkat enterprise pilihan. Ini adalah bahasa pengkodean serbaguna yang bagus karena dapat digunakan lintas platform (artinya itu sama mudahnya digunakan pada aplikasi smartphone seperti pada aplikasi desktop). Ini menyerupai C ++ dalam sintaks dan struktur sehingga mudah untuk diambil jika Anda sudah mengenal bahasa C. |

###### DBMS

Tabel 3 DBMS

|  |  |  |  |
| --- | --- | --- | --- |
|  | MySQL | MS SQL Server | Oracle |
| Primary database model | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) |
| Additional database models | [Document store](https://db-engines.com/en/article/Document+Stores) [Key-value store](https://db-engines.com/en/article/Key-value+Stores) | [Document store](https://db-engines.com/en/article/Document+Stores) [Key-value store](https://db-engines.com/en/article/Key-value+Stores) | [Document store](https://db-engines.com/en/article/Document+Stores) [Key-value store](https://db-engines.com/en/article/Key-value+Stores) |
| |  |  | | --- | --- | | [DB-Engines Ranking](https://db-engines.com/en/ranking) |  | | [Trend Chart](https://db-engines.com/en/ranking_trend/system/Microsoft+Access%3BMySQL%3BOracle) | | |  |  |  | | --- | --- | --- | | **Score** | 1298.83 | | | **Rank** | #2 | [**Overall**](https://db-engines.com/en/ranking) | |  | #2 | [**Relational DBMS**](https://db-engines.com/en/ranking/relational+dbms) | | **Score** 1210.32   |  |  |  | | --- | --- | --- | | **Rank** | #3 | [**Overall**](https://db-engines.com/en/ranking) | |  | #3 | [**Relational DBMS**](https://db-engines.com/en/ranking/relational+dbms) | | |  |  |  | | --- | --- | --- | | **Score** | 1348.80 | | | **Rank** | #1 | [**Overall**](https://db-engines.com/en/ranking) | |  | #1 | [**Relational DBMS**](https://db-engines.com/en/ranking/relational+dbms) | |
| Website | [www.mysql.com](https://www.mysql.com/) | [www.microsoft.com/­en-us/­sql-server](https://www.microsoft.com/en-us/sql-server/) | [www.oracle.com/­database/­index.html](https://www.oracle.com/database/index.html) |
| Technical documentation | [dev.mysql.com/­doc](https://dev.mysql.com/doc/) | [docs.microsoft.com/­en-ie/­sql/­sql-server/­sql-server-technical-documentation](https://docs.microsoft.com/en-ie/sql/sql-server/sql-server-technical-documentation) | [docs.oracle.com/­en/­database](https://docs.oracle.com/en/database/) |
| Developer | Oracle | Microsoft | Oracle |
| Initial release | 1995 | 1989 | 1980 |
| Current release | 5.7.20, October 2017 | SQL Server 2017, October 2017 | 12 Release 2 (12.2.0.1), March 2017 |
| License | Open Source | commercial | commercial |
| Cloud-based | no | no | no |
| Implementation language | C and C++ | C++ | C and C++ |
| Server operating systems | FreeBSD Linux OS X Solaris Windows | Linux Windows | AIX HP-UX Linux OS X Solaris Windows z/OS |
| Data scheme | yes | yes | yes |
| Typing | yes | yes | yes |
| XML support | yes | yes | Yes |
| Secondary indexes | yes | yes | Yes |
| SQL | yes | yes | Yes |
| APIs and other access methods | ADO.NET JDBC ODBC | OLE DB Tabular Data Stream (TDS) ADO.NET JDBC ODBC | ODP.NET Oracle Call Interface (OCI) JDBC ODBC |
| Supported programming languages | Ada C C# C++ D Delphi Eiffel Erlang Haskell Java JavaScript (Node.js) Objective-C OCaml Perl PHP Python Ruby Scheme Tcl | C++ Delphi Go Java JavaScript (Node.js) PHP Python R Ruby Visual Basic | C C# C++ Clojure Cobol Delphi Eiffel Erlang Fortran Groovy Haskell Java JavaScript Lisp Objective C OCaml Perl PHP Python R Ruby Scala Tcl Visual Basic |
| Server-side scripts | yes | Transact SQL and .NET languages | PL/SQL |
| Triggers | yes | yes | Yes |
| Partitioning methods | horizontal partitioning, sharding with MySQL Cluster or MySQL Fabric | tables can be distributed across several files (horizontal partitioning); sharding through federation | horizontal partitioning |
| Replication methods | Master-master replication Master-slave replication | yes, but depending on the SQL-Server Edition | Master-master replication Master-slave replication |
| MapReduce | No | No | No |
| Consistency concepts | Immediate Consistency | Immediate Consistency | Immediate Consistency |
| Foreign keys | yes | Yes | yes |
| Transaction concepts | ACID | ACID | ACID |
| Concurrency | yes | yes | yes |
| Durability | yes | yes | yes |
| In-memory capabilities | yes | yes | yes |
| User concepts | Users with fine-grained authorization concept | fine grained access rights according to SQL-standard | fine grained access rights according to SQL-standard |

# Test Plan

## UAT

User Acceptance Test (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa software yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (testing) sudah bisa dianggap memenuhi kebutuhan dari pengguna.

# DAFTAR PUSTAKA

<https://betha.wordpress.com/2006/05/01/user-acceptance-test/>

<https://db-engines.com/en/system/Microsoft+SQL+Server>

<https://db-engines.com/en/system/MySQL>

https://db-engines.com/en/system/Oracle