

# **Sistem Informasi Inventaris dan**

## **Peminjaman Alat**

Aplikasi berbasis web menggunakan Framework Laravel untuk mengelola inventaris alat, proses peminjaman oleh user, serta verifikasi pengembalian dan perhitungan denda otomatis oleh petugas/admin.

```
class User extends Authenticatable
{
    /**
     * The attributes that should be mutated for serialization.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }

    public function dashboardUrl()
    {
        //ret match($this->role => role.ds)
        return match($this->role) {
            'admin' => route('admin.dashboard'),
            'petugas' => route('officer.dashboard'),
            default => route('borrower.dashboard'),
        };
    }

    public function loans()
    {
        return $this->hasMany(Loan::class, 'borrower_id');
    }
}
```

```
public function update(Request $request, $id)
{
    $validated = $request->validate($rules);

    $user = User::where('username', $validated['username'])
        ->where('email', $validated['email']);

    if (isset($validated['password'])) {
        $user->password = bcrypt($validated['password']);
    }

    if (isset($validated['role'])) {
        $user->role = $validated['role'];
    }

    $user->save();

    return redirect()->route('users.index')->with('success', 'User berhasil diupdate');
}

public function destroy(Request $request, $id)
{
    $user = User::findOrFail($id);

    if ($user->id === auth()->id()) {
        return back()->with('error', 'Tidak bisa menghapus akun sendiri.');
    }

    $hasActiveLoans = $user->loans()->exists();

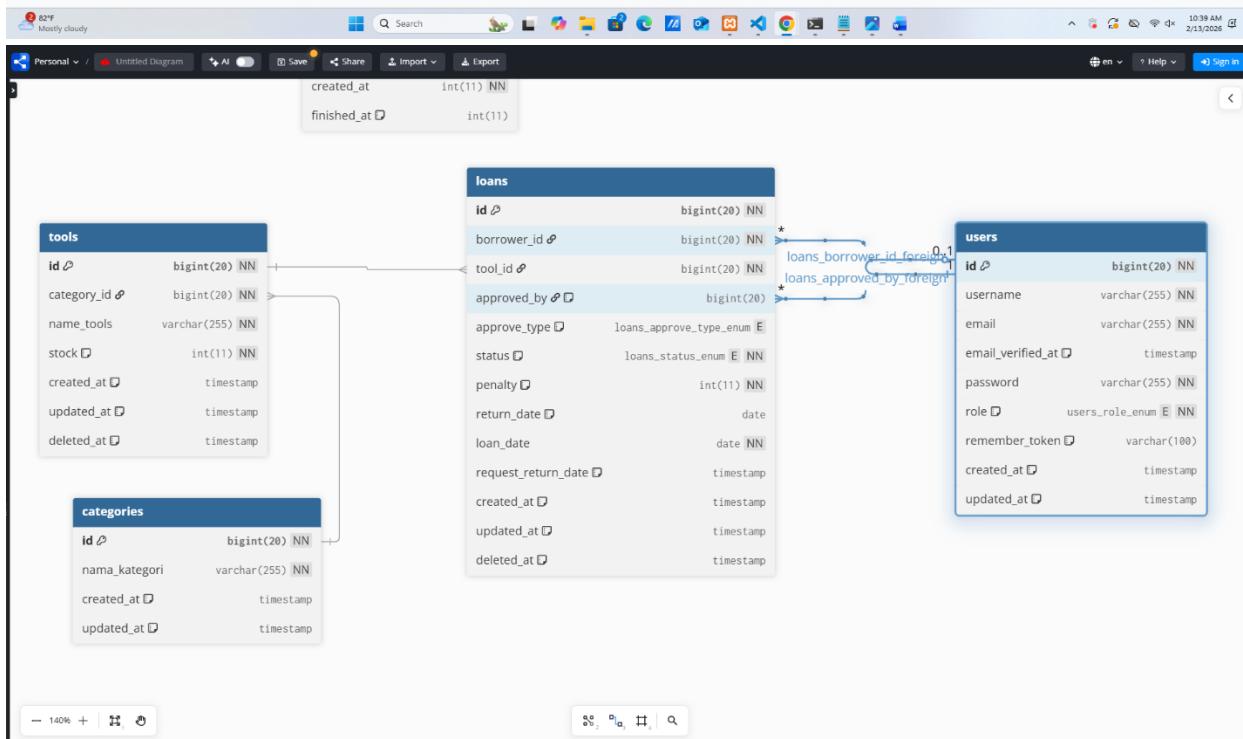
    if ($hasActiveLoans) {
        return back()->with('error', 'User tidak bisa dihapus karena memiliki riwayat atau transaksi pinjaman!');
    }

    // Jika lolos pengecekan, hapus user
    $user->delete();

    return back()->with('success', 'User berhasil dihapus');
}
```

Screenshot of a web application titled "PinjamAlat". The URL is "localhost/peminjaman\_lar/public/users". The top navigation bar includes links for Category, Tool, User, Loan, Logs, and Logout (admin). A pink message box at the top states: "User tidak bisa dihapus karena memiliki riwayat atau transaksi peminjaman!". Below this is a table titled "Users List" showing four users:

Name	Email	Role	Action
talon.schoen	ybrekke@example.org	BORROWER	
hirthe.heber	kreiger.declan@example.com	BORROWER	
rodriguez.isobel	runolfsson.clemmie@example.net	BORROWER	
brenden39	willy.pouros@example.net	BORROWER	



Screenshot of a code editor showing two files: `LoanController.php` and `UserController.php`.

```

LoanController.php (Line 18):
18 class LoanController extends Controller
19 {
20     public function index()
21     {
22         return view('penitjaman.index', compact('tools'));
23     }
24 }

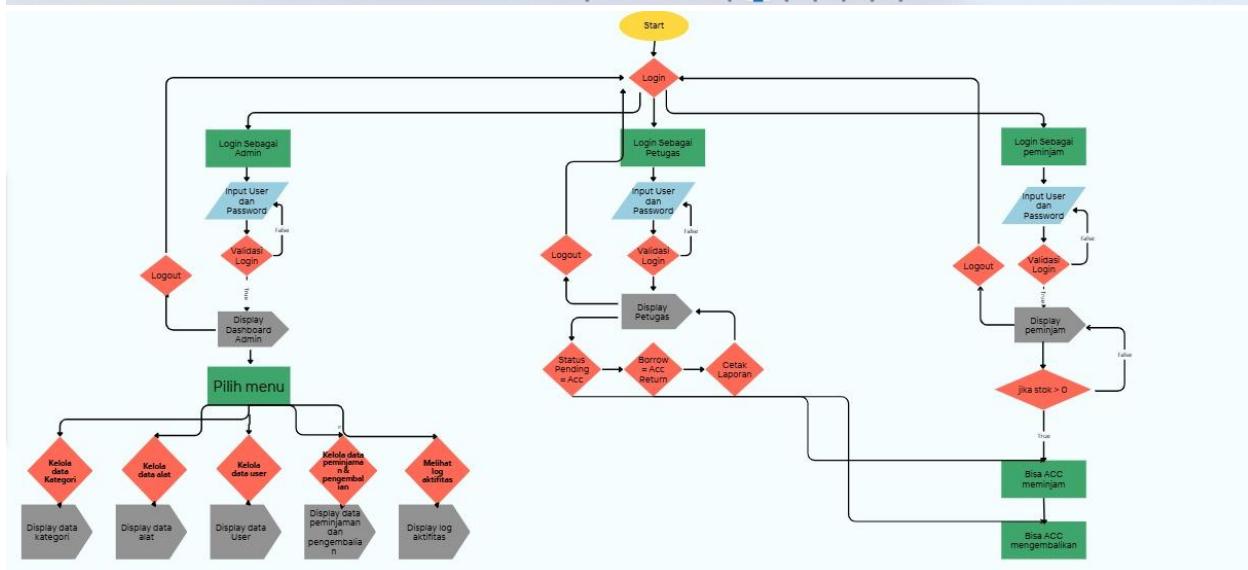
UserController.php (Line 69):
69     public function show($id)
70     {
71         /**
72          * Show the form for editing the specified resource.
73          */
74
75         public function edit($id, Request $request)
76         {
77             // valid tool id need
78             $request->validate(['tool_id' => 'required']);
79
80             // start store
81             DB::beginTransaction();
82             // try the case
83             try {
84                 // get tool id
85                 $tool = tool::findOrFail($request->tool_id);
86                 // ret
87                 if ($tool->stock <= 0) return back()->with('error', 'Stok habis');
88
89                 // logic get ID n usn
90                 if ($request->has('user_id')) {
91                     // admin input
92                     $borrower = User::findOrFail($request->user_id);
93                     $borrowerId = $borrower->id;
94                     $borrowerName = $borrower->username;
95                 } else {
96                     // borrower login
97                     $user = User::Auth::user();
98                     $borrowerId = $user->id;
99                     $borrowerName = $user->username;
100                }
101
102                // tool->decr(stock)
103                // $tool->decrement('stock');
104
105                // create loan([loan row])
106                // $loan =
107
108            } catch (\Exception $e) {
109                // tool->increment('stock');
110                // $tool->increment('stock');
111            }
112        }
113    }

```

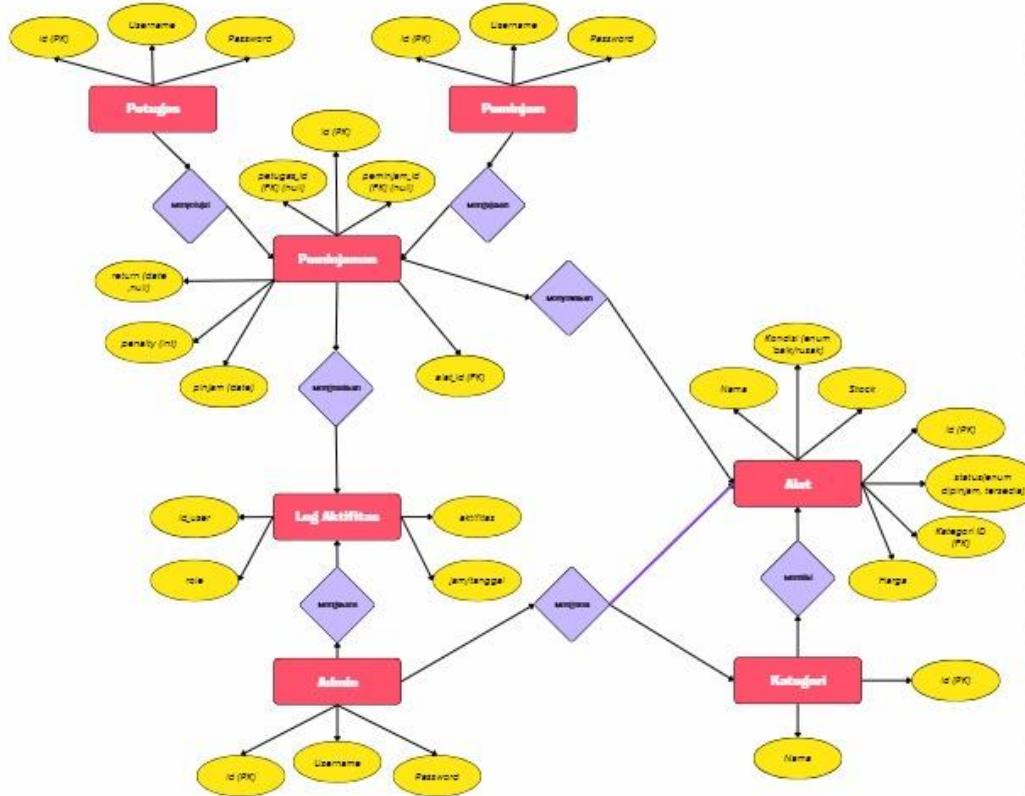
```

UserController.php (Line 69):
69     public function show($id)
70     {
71         /**
72          * Show the form for editing the specified resource.
73          */
74
75         public function edit($id, Request $request)
76         {
77             // lower var
78             $modelname = $this->getModel($role);
79             $user = User::findOrFail($id);
80
81             return view('admin.users.edit', compact('user'));
82         }
83
84         public function update(Request $request, $id)
85         {
86             $user = User::findOrFail($id);
87
88             if ($user->role === 'admin' && $request->role !== 'admin') {
89                 return back()->with('error', 'Role Admin tidak boleh diubah menjadi');
90             }
91
92             // rules dasar
93             $rules = [
94                 'username' -> 'required|unique:users,username,' . $id,
95                 'email' -> 'required|email|unique:users,email,' . $id,
96             ];
97
98             // password optional
99             if ($request->filled('password')) {
100                 $rules['password'] = 'string:min:8|confirmed';
101             }
102
103             // role hanya dilinkan untuk non-admin
104             if ($user->role != 'admin') {
105                 $rules['role'] = 'required|in:borrower,officer,admin';
106             }
107
108             $validated = $request->validate($rules);
109
110             $user->username = $validated['username'];
111             $user->email = $validated['email'];
112         }
113     }

```



## ERD



Laravel Internal Server Error

Illuminate\Database\QueryException

vendor\laravel\framework\src\Illuminate\Database\Connection.php:831

SQLSTATE[23000]: Integrity constraint violation: 1451 Cannot delete or update a parent row: a foreign key constraint fails ('db\_ryuu`.`loans', CONSTRAINT `loans\_borrower\_id\_foreign` FOREIGN KEY (`borrower\_id`) REFERENCES `users` (`id`)) (Connection: mysql, Host: 127.0.0.1, Port: 3306, Database: db\_ryuu, SQL: delete from `users` where `id` = 3)

LARAVEL 12.46.0 PHP 8.2.12 UNHANDLED CODE 23000

500 DELETE http://localhost/peminjaman\_lar/public/users/3?role=borrower

Exception trace

9 vendor frames

App\Http\Controllers\UserController->destroy(object(Illuminate\Http\Request), string) app\Http\Controllers\UserController.php:139

```

134     // simp protect fail
135     if ($user->id === auth()->id()) {
136         return back()->with('error', 'tidak bisa menghapus akun sendiri');
137     }
  
```

Rainy days ahead 62°F