**Laporan Final Project**

**Perancangan Perangkat Lunak**

**Kelas: B**



**Kelompok F**

**Nama Anggota**

**Patrick Sungkharisma 05111740000041**

**Jonathan Alphabert S 05111740000053**

**Fandi Pranata Jaya 05111740000056**

**Dosen:**

**Adhatus Solichah Ahmadiyah, S.Kom, M.Sc**

**Departemen Teknik Infomatika**

**Fakultas Teknologi Informasi dan Komunikasi**

**Institut Teknologi Sepuluh Nopember (ITS)**

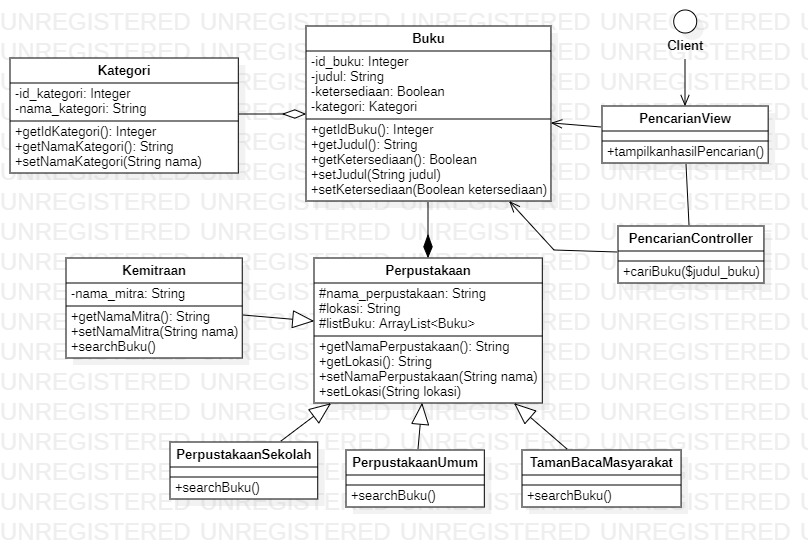
**Surabaya**

**2019**

1. **Deskripsi Sistem**

Sistem informasi yang dianalisa adalah sistem informasi Perpustakaan Pemerintah Kota Surabaya. Pada sistem tersebut, hal yang akan kami modifikasi adalah dalam hal request pencarian buku dan request peminjaman buku. Pada sistem tersebut, terdapat beberapa perpustakaan yang berperan sebagai agent. Awalnya setiap perpustakaan memiliki buku yang dapat dilihat dan dicari oleh client melalui request yang dilakukan terhadap website <https://dispusip.surabaya.go.id/dils/>. Kemudian kami menambahkan fitur peminjaman buku yang sudah dicari sebelumnya ke salah satu perpustakaan.

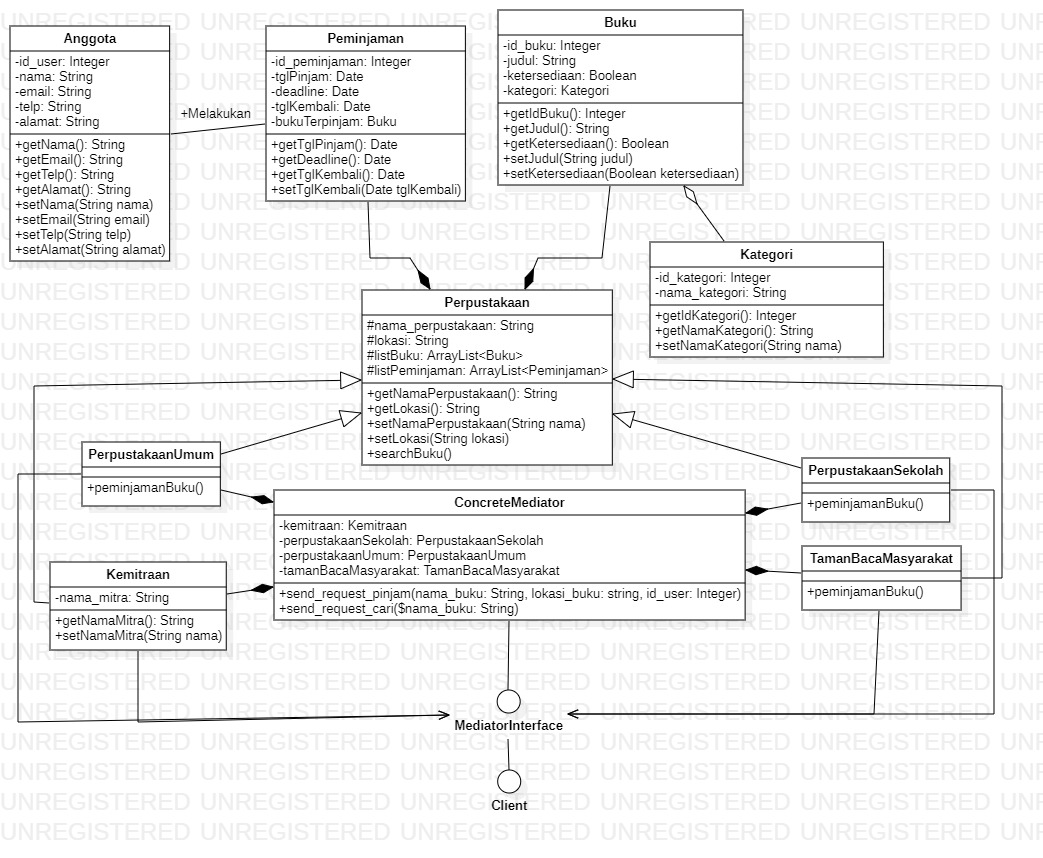
1. **Class Diagram Tanpa Design Pattern**



Class diagram diatas diterapkan dengan menggunakan arsitektur MVC (Model, View, Controller). Terdapat fitur utama yaitu fitur pencarian buku. Client yang ingin menggunakan fitur tersebut akan melalui tampilan yang diatur oleh view dari fitur yang pada class diagram ini dinamakan PencarianView. Model yang digunakan pada fitur pencarian adalah Buku dan Perpustakaan. Query – query yang dibutuhkan pada fitur tersebut akan diatur oleh PencarianController.

Class diagram diatas belum menerapkan Design Pattern, semua class terpisah – pisah dan menjalankan tugasnya masing – masing.

1. **Class Diagram Dengan Design Pattern**



Untuk Class Diagram ini, kami menerapkan Design Pattern Mediator. Peran Design Pattern Mediator dalam sistem ini adalah untuk mengatur lalu lintas request peminjaman yang terjadi dari beberapa agent – agent perpustakaan pada saat website tersebut diakses oleh banyak user sehingga tidak ada request peminjaman yang bertabrakan. Pada sistem ini client mengakses sistem melalui MediatorInterface yang nantinya akan dihubungkan dengan class ConcreteMediator yang bertugas untuk mengatur lalu lintas request pencarian buku dan terutama request peminjaman buku dari client.

1. **Implementasi**
2. Class Anggota

<?php

class Anggota

{

    private $id\_user, $nama, $email, $telp, $alamat;

    public function \_\_get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama;

*case* 'email':

*return* *$this*->email;

*case* 'telp':

*return* *$this*->telp;

*case* 'alamat':

*return* *$this*->alamat;

        }

    }

    public function \_\_set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama = $value;

*break*;

*case* 'email':

*$this*->email = $value;

*break*;

*case* 'telp':

*$this*->telp = $value;

*break*;

*case* 'alamat':

*$this*->alamat = $value;

*break*;

        }

    }

}

1. Class Buku

<?php

class Buku

{

    private $id\_buku, $judul, $ketersediaan, $kategori;

    public function \_\_construct($judul, $kategori)

    {

*$this*->judul=$judul;

*$this*->kategori=$kategori;

*$this*->ketersediaan=true;

    }

    public function get($choice)

    {

*switch*($choice)

        {

*case* 'judul':

*return* *$this*->judul;

*case* 'ketersediaan':

*return* *$this*->ketersediaan;

        }

    }

    public function set($choice, $value)

    {

*switch*($choice)

        {

*case* 'judul':

*$this*->judul = $value;

*break*;

*case* 'ketersediaan':

*$this*->ketersediaan = $value;

*break*;

        }

    }

}

1. Class Kategori

<?php

class Kategori

{

    private $id\_kategori, $nama\_kategori;

    public function \_\_get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama\_kategori;

        }

    }

    public function \_\_set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama\_kategori = $value;

*break*;

        }

    }

}

1. Class Kemitraan

<?php

class Kemitraan extends Perpustakaan

{

    private $mitra;

    public function \_\_construct($nama, $lokasi, $mitra)

    {

*$this*->nama = $nama;

*$this*->lokasi = $lokasi;

*$this*->listbuku = [];

*$this*->peminjaman = [];

*$this*->jenis = "Kemitraan";

*$this*->mitra=$mitra;

    }

    public function get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama;

*case* 'lokasi':

*return* *$this*->lokasi;

*case* 'jenis':

*return* *$this*->jenis;

        }

    }

    public function set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama = $value;

*break*;

*case* 'lokasi':

*$this*->lokasi = $value;

*break*;

*case* 'buku':

                array\_push(*$this*->listbuku, $value);

*break*;

        }

    }

    public function cari\_buku($nama\_buku)

    {

        echo nl2br("Terhubung dengan perpustakaan $this->nama\n");

*// print(newline);*

        echo nl2br("Mencari buku $nama\_buku pada perpustakaan $this->nama\n");

        $found = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

                $found = 1;

            }

        }

        echo nl2br("Mengirim hasil pencarian dari perpustakaan $this->nama\n");

*return* $found;

    }

}

1. Class Mediator

<?php

class Mediator

{

    private $daftar\_perpustakaan;                                   *# Array List Perpustakaan yang masuk ke sistem*

    public function \_\_construct($list)

    {

*$this*->daftar\_perpustakaan = $list;

    }

    public function send\_request\_pinjam($nama\_buku, $lokasi\_buku, $id\_user)

    {

        echo nl2br("\nMediator menerima request peminjaman $nama\_buku pada perpustakaan $lokasi\_buku\n");

*foreach*(*$this*->daftar\_perpustakaan as $perpus)

        {

*if*($lokasi\_buku == $perpus->nama)

            {

                echo nl2br("Mediator: Menghubungkan pada perpustakaan $perpus->nama \n");

                $success = $perpus->peminjamanBuku($nama\_buku, $id\_user);

                echo nl2br("Mediator: Menerima hasil peminjaman dari perpustakaan $perpus->nama\n");

*if*($success)

                {

                    echo nl2br("Mediator : Peminjaman Sukses\n");

                }

*else*

                {

                    echo nl2br("Mediator : Peminjaman Tidak Berhasil dikarena buku yang anda minta tidak tersedia pada perpustakaan ini.\n");

                }

            }

        }

    }

    public function send\_request\_cari($nama\_buku)

    {

        echo nl2br("Mediator menerima request untuk mencari buku dengan judul $nama\_buku\n\n");

        $jumlah\_perpus = 0;

        $list\_perpus = [];

*foreach*(*$this*->daftar\_perpustakaan as $perpus)

        {

            echo nl2br("Mediator: Menghubungkan pada perpustakaan $perpus->nama \n");

            $found = $perpus->cari\_buku($nama\_buku);

            echo nl2br("Mediator: Menerima hasil pencarian dari perpustakaan $perpus->nama\n\n");

*if*($found)

            {

                array\_push($list\_perpus, $perpus->nama);

                $jumlah\_perpus++;

            }

        }

*if*($jumlah\_perpus==0)

        {

            echo nl2br("Mediator    : Buku tidak ditemukan di perpustakaan manapun.\n");

        }

*else*{

            print("Mediator     : Buku yang anda cari ada di\n");

*foreach*($list\_perpus as $p)

            {

                echo nl2br("$p\n");

            }

        }

    }

}

1. Class Peminjaman

<?php

class Peminjaman

{

    private $id\_peminjaman, $id\_user, $tglPinjam, $deadline, $tglKembali, $bukuTerpinjam;

    public function \_\_constructor($nama\_buku, $id\_user)

    {

*$this*->bukuTerpinjam = $nama\_buku;

*$this*->id\_user = $id\_user;

*$this*->tglPinjam = TIMESTAMP;

*$this*->deadline = *$this*->tglPinjam;

        date\_add(*$this*->deadline , date\_interval\_create\_from\_date\_string("7 days"));

    }

    public function \_\_get($choice)

    {

*switch*($choice)

        {

*case* 'tglPinjam':

*return* *$this*->tglPinjam;

*case* 'deadline':

*return* *$this*->deadline;

*case* 'tglKembali':

*return* *$this*->tglKembali;

        }

    }

    public function setTanggalKembali()

    {

*$this*->tglKembali=TIMESTAMP;

    }

}

1. Class Perpustakaan

<?php

class Perpustakaan

{

    protected $nama, $lokasi, $listbuku, $listpeminjaman, $jenis;

    public function \_\_constructor($nama, $lokasi)

    {

*$this*->nama = $nama;

*$this*->lokasi = $lokasi;

*$this*->listbuku = [];

*$this*->peminjaman = [];

    }

    public function \_\_get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama;

*case* 'lokasi':

*return* *$this*->lokasi;

*case* 'jenis':

*return* *$this*->jenis;

        }

    }

    public function \_\_set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama = $value;

*break*;

*case* 'lokasi':

*$this*->lokasi = $value;

*break*;

        }

    }

    public function cari\_buku($nama\_buku)

    {

        print("Terhubung dengan perpustakaan " + *$this*->nama);

        print("Mencari buku" + $nama\_buku + "pada perpustakaan " + *$this*->nama);

        $found = 0;

*// foreach($listbuku as $buku)*

*// {*

*//     if($buku->nama == $nama\_buku)*

*//     {*

*//         $found = 1;*

*//     }*

*// }*

*for*($i=0; $i<length(listbuku); $i++)

        {

*if*($buku->nama == $nama\_buku)

            {

                $found = 1;

            }

        }

        print("Mengirim hasil pencarian dari perpustakaan " + *$this*->nama);

*return* $found;

    }

}

1. Class PerpustakaanSekolah

<?php

class PerpustakaanSekolah extends Perpustakaan

{

    public function \_\_construct($nama, $lokasi)

    {

*$this*->nama = $nama;

*$this*->lokasi = $lokasi;

*$this*->listbuku = [];

*$this*->peminjaman = [];

*$this*->jenis = "Perpustakaan Sekolah";

    }

    public function get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama;

*case* 'lokasi':

*return* *$this*->lokasi;

*case* 'jenis':

*return* *$this*->jenis;

        }

    }

    public function set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama = $value;

*break*;

*case* 'lokasi':

*$this*->lokasi = $value;

*break*;

*case* 'buku':

                array\_push(*$this*->listbuku, $value);

*break*;

        }

    }

    public function peminjamanBuku($nama\_buku, $id\_user)

    {

        echo nl2br("$this->jenis :Terhubung dengan perpustakaan $this->nama\n");

        echo nl2br("$this->jenis :Mencari ketersediaan buku $nama\_buku pada perpustakaan $this->nama\n");

        $tersedia = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

*if*($buku->get('ketersediaan'))             *# Buku tersedia*

                {

                    $temp = new Peminjaman($nama\_buku, $id\_user);

                    array\_push($listPeminjaman, $temp);

                    $tersedia = 1;

*break*;

                }

            }

        }

        echo nl2br("$this->jenis :Mengirim hasil peminjaman dari perpustakaan $this->nama\n");

*return* $tersedia;

    }

    public function cari\_buku($nama\_buku)

    {

        echo nl2br("$this->jenis : Terhubung dengan perpustakaan $this->nama\n");

*// print(newline);*

        echo nl2br("$this->jenis : Mencari buku $nama\_buku pada perpustakaan $this->nama\n");

        $found = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

                $found = 1;

            }

        }

        echo nl2br("$this->jenis : Mengirim hasil pencarian dari perpustakaan $this->nama\n");

*return* $found;

    }

}

1. Class PerpustakaanUmum

<?php

class PerpustakaanUmum extends Perpustakaan

{

    public function \_\_construct($nama, $lokasi)

    {

*$this*->nama = $nama;

*$this*->lokasi = $lokasi;

*$this*->listbuku = [];

*$this*->peminjaman = [];

*$this*->jenis = "Perpustakaan Umum";

    }

    public function get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama;

*case* 'lokasi':

*return* *$this*->lokasi;

*case* 'jenis':

*return* *$this*->jenis;

        }

    }

    public function set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama = $value;

*break*;

*case* 'lokasi':

*$this*->lokasi = $value;

*break*;

*case* 'buku':

                array\_push(*$this*->listbuku, $value);

*break*;

        }

    }

    public function peminjamanBuku($nama\_buku, $id\_user)

    {

        echo nl2br("$this->jenis : Terhubung dengan perpustakaan $this->nama\n");

        echo nl2br("$this->jenis : Mencari ketersediaan buku $nama\_buku pada perpustakaan $this->nama\n");

        $tersedia = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

*if*($buku->get('ketersediaan'))             *# Buku tersedia*

                {

                    $temp = new Peminjaman($nama\_buku, $id\_user);

                    array\_push($listPeminjaman, $temp);

                    $tersedia = 1;

*break*;

                }

            }

        }

        echo nl2br("$this->jenis : Mengirim hasil peminjaman dari perpustakaan $this->nama\n");

*return* $tersedia;

    }

    public function cari\_buku($nama\_buku)

    {

        echo nl2br("$this->jenis : Terhubung dengan perpustakaan $this->nama\n");

*// print(newline);*

        echo nl2br("$this->jenis : Mencari buku $nama\_buku pada perpustakaan $this->nama\n");

        $found = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

                $found = 1;

            }

        }

        echo nl2br("$this->jenis : Mengirim hasil pencarian dari perpustakaan $this->nama\n");

*return* $found;

    }

}

1. Class TamanBacaMasyarakat

<?php

class TamanBacaMasyarakat extends Perpustakaan

{

    public function \_\_construct($nama, $lokasi)

    {

*$this*->nama = $nama;

*$this*->lokasi = $lokasi;

*$this*->listbuku = [];

*$this*->peminjaman = [];

*$this*->jenis = "Taman Baca Masyarakat";

    }

    public function get($choice)

    {

*switch*($choice)

        {

*case* 'nama':

*return* *$this*->nama;

*case* 'lokasi':

*return* *$this*->lokasi;

*case* 'jenis':

*return* *$this*->jenis;

        }

    }

    public function set($choice, $value)

    {

*switch*($choice)

        {

*case* 'nama':

*$this*->nama = $value;

*break*;

*case* 'lokasi':

*$this*->lokasi = $value;

*break*;

*case* 'buku':

                array\_push(*$this*->listbuku, $value);

*break*;

        }

    }

    public function peminjamanBuku($nama\_buku, $id\_user)

    {

        echo nl2br("Terhubung dengan perpustakaan $this->nama\n");

        echo nl2br("Mencari ketersediaan buku $nama\_buku pada perpustakaan $this->nama\n");

        $tersedia = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

*if*($buku->get('ketersediaan'))             *# Buku tersedia*

                {

                    $temp = new Peminjaman($nama\_buku, $id\_user);

                    array\_push($listPeminjaman, $temp);

                    $tersedia = 1;

*break*;

                }

            }

        }

        echo nl2br("Mengirim hasil peminjaman dari perpustakaan $this->nama\n");

*return* $tersedia;

    }

    public function cari\_buku($nama\_buku)

    {

        echo nl2br("Terhubung dengan perpustakaan $this->nama\n");

*// print(newline);*

        echo nl2br("Mencari buku $nama\_buku pada perpustakaan $this->nama\n");

        $found = 0;

*foreach*(*$this*->listbuku as $buku)

        {

*if*($buku->get('judul') == $nama\_buku)

            {

                $found = 1;

            }

        }

        echo nl2br("Mengirim hasil pencarian dari perpustakaan $this->nama\n");

*return* $found;

    }

}

1. Main/Index

<?php

*include* 'perpustakaan.php';

*include* 'perpustakaansekolah.php';

*include* 'buku.php';

*include* 'peminjaman.php';

*include* 'anggota.php';

*include* 'kategori.php';

*include* 'kemitraan.php';

*include* 'mediator.php';

*include* 'perpustakaanUmum.php';

*include* 'tamanbacamasyarakat.php';

    error\_reporting(E\_ERROR | E\_PARSE);

    $buku1 = new Buku("Belajar HTML", "Komputer");

    $buku2 = new Buku("Belajar CSS", "Komputer");

    $buku3 = new Buku("Belajar JAVASCRIPT", "Komputer");

    $buku4 = new Buku("Belajar PHP", "Komputer");

    $buku5 = new Buku("Belajar LARAVEL", "Komputer");

    $buku6 = new Buku("Belajar React.js", "Komputer");

    $perpustakaan = new PerpustakaanSekolah("SMP N 2", "Jl. Darmo");

    $perpustakaan2 = new PerpustakaanUmum("Kota Surabaya", "Jl. Pahlawan");

    $perpustakaan3 = new PerpustakaanSekolah("SMA N 2", "Jl. Biliton");

    $perpustakaan->set('buku', $buku1);

    $perpustakaan->set('buku', $buku3);

    $perpustakaan->set('buku', $buku5);

    $perpustakaan2->set('buku', $buku2);

    $perpustakaan2->set('buku', $buku4);

    $perpustakaan2->set('buku', $buku6);

    $perpustakaan3->set('buku', $buku6);

    $perpustakaan3->set('buku', $buku3);

    $list\_perpustakaan = [];

    array\_push($list\_perpustakaan, $perpustakaan, $perpustakaan2, $perpustakaan3);

    $mediator = new Mediator($list\_perpustakaan);

    $mediator->send\_request\_cari("Belajar HTML");

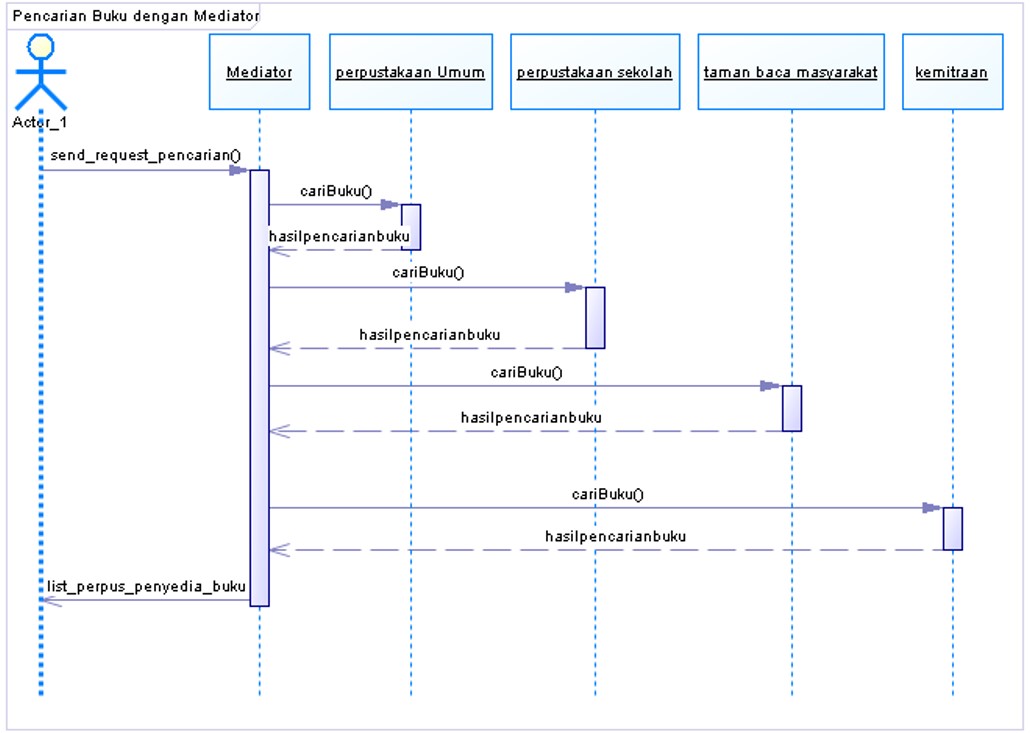
    $mediator->send\_request\_pinjam("Belajar HTML", "SMP N 2", 0);

1. Hasil setelah dijalankan

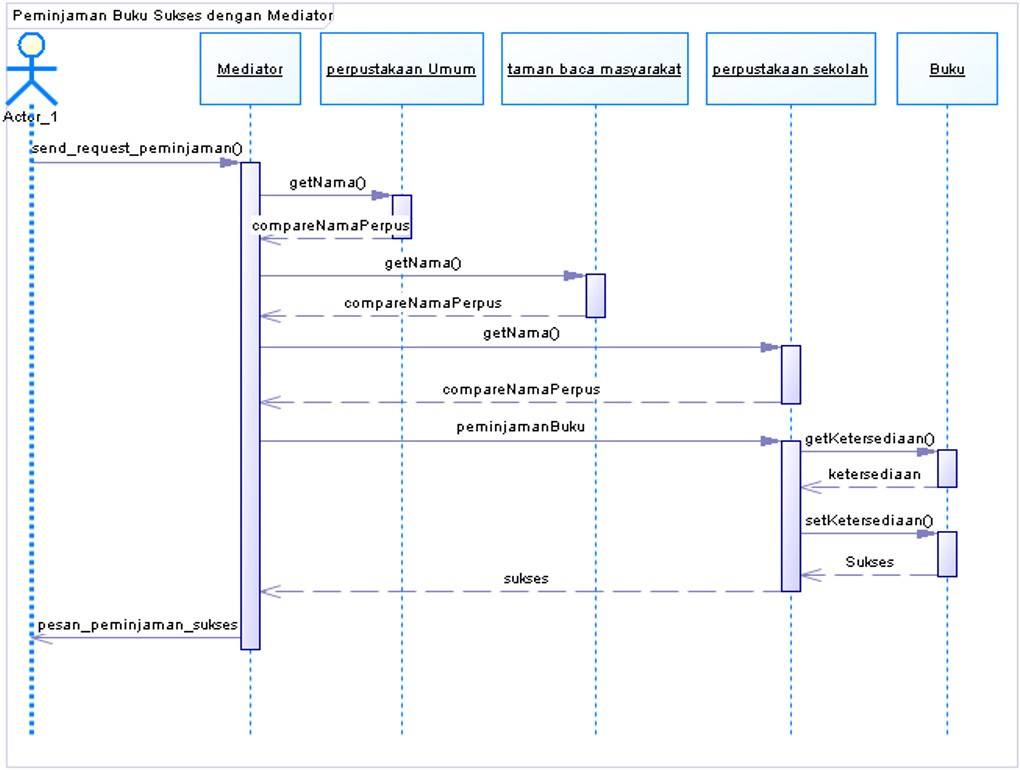


Pada gambar diatas merupakan ilustrasi alur dari setiap proses peminjaman dan juga pencarian dengan design pattern Mediator. Pada gambar tersebut, terlihat bahwa setiap kali hendak melakukan request, mediator selalu membuat koneksi terhadap perpustakaan yang terdapat pada sistem. Didalam setiap perpustakaan barulah dilakukan proses peminjaman atau bahkan saat proses pencarian buku.

1. Sequence Diagram
2. Pencarian Buku Dengan Mediator



1. Peminjaman Buku Sukses Dengan Mediator



1. Peminjaman Buku Gagal Dengan Mediator

