

**BUKU TUTORIAL APLIKASI**

**“RANCANG BANGUN APLIKASI E-OFFICE BERBASIS WEB**

**DENGAN MENGGUNAKAN METODE RAD”**

Buku ini dibuat untuk memenuhi persyaratan kelulusan  
matakuliah Program Internship I



**Dibuat Oleh,**  
**1.16.4.023 RAYMANA APRILIAN**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA**  
**POLITEKNIK POS INDONESIA**  
**BANDUNG**  
**2019**

## **KATA PENGANTAR**

---

Assalamualaikum warahmatullahi wabarakatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan Internship I ini, tanpa pertolongan-Nya mungkin penulis tidak akan sanggup menyelesaikannya dengan baik. Shalawat dan salam semoga terlimpah curahkan kepada Nabi Muhammad SAW beserta sahabat dan keluarga Beliau.

Laporan ini disusun untuk memenuhi kelulusan matakuliah Internship I pada Program Studi DIV Teknik Informatika. Proses Internship I ini juga tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, pada kata pengantar ini penulis menyampaikan terimakasih kepada :

1. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika Tahun Akademik 2019/2020;
2. Nisa Hanum Harani, S.Kom, M.T. selaku Koordinator Internship I Tahun Akademik 2019/2020;
3. Roni Habibi, S.Kom., M.T. selaku Pembimbing Internal dan Penguji Utama dalam penyusunan laporan Internship I ini;
4. Lia Marlian Sulaeman, A.Md. selaku Pembimbing Eksternal dalam penyusunan buku Internship I ini;
5. Teman-teman penulis yang telah banyak membantu dalam menyelesaikan tugas ini.

Penulis telah membuat laporan ini dengan sebaik-baiknya, diharapkan memberikan kritik dan saran dari semua pihak yang bersifat membangun, terimakasih.

Bandung, 21 Januari 2020

Penulis

## DAFTAR ISI

---

KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
BAB I.....	1
1.1    PENGENALAN PHP, CODEIGNITER, DAN DATABASE.....	1
1.1.1    Pengantar PHP (Hypertext Preprocessor).....	1
1.1.2    Pengantar CodeIgniter .....	9
1.1.3    Pengantar Database.....	19
BAB II.....	27
2.1    PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN .....	27
2.1.1    Tools Yang Digunakan .....	27
2.1.2    Bahasa Pemograman .....	42
BAB III .....	44
3.1    INSTALASI TOOLS YANG DIGUNAKAN .....	44
3.1.1    Tools Yang Digunakan .....	44
BAB IV .....	65
4.1    PENGENALAN DAN PENJELASAN METODE RAD (RAPID APPLICATION DEVELOPMENT) .....	65
4.1.1    Pengantar Metode RAD .....	65
4.2.1    Diagram Alur Metodologi Penelitian .....	71
Tahapan – tahapan Diagram Alur Metodologi Penelitian.....	71
4.2.2    Pengumpulan Kebutuhan .....	71
4.2.3    Membangun Prototyping .....	72
4.2.4    Evaluasi <i>Prototyping</i> .....	72
4.2.5    Implementasi.....	72
4.2.6    Pengujian.....	73
4.2.7    Kesimpulan dan Saran .....	73
BAB V .....	74

<b>5.1 ANALISIS DAN PERANCANGAN SISTEM .....</b>	74
<b>5.2 Perancangan .....</b>	83
<b>5.2.1 Perancangan UML .....</b>	84
5.2.2 Class Diagram.....	95
<b>5.2.3 Sequence Diagram .....</b>	99
5.2.4 Collaboration Diagram .....	105
5.2.5 Activity Diagram.....	111
5.2.6 Statechart Diagram .....	119
5.2.7 Component Diagram .....	129
5.2.8 Deployment Diagram .....	136
5.2.9 Perancangan Antar Muka .....	147
<b>BAB VI.....</b>	2
<b>6.1 PENGUJIAN DENGAN BLACKBOX .....</b>	2
<b>6.1.1 PENGENALAN BLACKBOX .....</b>	2
6.1 Lingkungan Implementasi..... <b>Error! Bookmark not defined.</b>	
6.1.1 Kebutuhan Perangkat Lunak.....	15
6.1.2 Kebutuhan Perangkat Keras.....	15
1.2 6.2 Pengujian Dan Hasil Pengujian.....	16
1. 6.2.1 Identifikasi dan Rencana Pengujian .....	16
2. 6.2.2 Deskripsi dan Hasil Uji .....	17
<b>BAB VII.....</b>	21
<b>7.1 TUTORIAL PENGGUNAAN APLIKASI E-OFFICE .....</b>	21

## BAB I

---

### 1.1 PENGENALAN PHP, CODEIGNITER, DAN DATABASE

Php (Hypertext Preprocessor), Code Igniter, dan Database adalah istilah dalam dunia teknologi informasi yang sangat populer saat ini. Pada buku ini penulis akan memaparkan materi tentang teknologi tersebut.

#### 1.1.1 Pengantar PHP (Hypertext Preprocessor)

##### **Definisi PHP (*Hypertext Preprocessor*)**

Bahasa pemrograman PHP biasanya tidak digunakan pada keseluruhan pengembangan website, melainkan dikombinasikan dengan beberapa bahasa pemrograman lain. Misalnya saja untuk mengatur tampilan, layout, dan berbagai macam menu menggunakan CSS. Selain itu, terdapat juga beberapa framework PHP; Laravel, Phalcon, Codigniter, Symfoni yang saat ini banyak tersedia di internet untuk memudahkan proses pengembangan website menggunakan bahasa pemrograman tersebut.

Agar Anda lebih mengenai bahasa pemrograman PHP dan CSS, kita akan membahas terlebih dahulu apa itu PHP dan apa fungsi PHP di dalam proses pengembangan website.

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Bahasa pemrograman ini menggunakan sistem server-side. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya script/program tersebut akan dijalankan/diproses oleh server. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari.

Semenjak PHP menjadi bahasa pemrograman yang open source, pengembang tidak perlu menunggu sampai dengan update terbaru rilis.

Pengguna PHP akan lebih baik jika menggunakan versi terbaru. Sehingga jika ada rilis terbaru Anda harus menyesuaikan sistem Anda dengan versi PHP yang paling baru. Meskipun harus menggunakan versi terbaru, biaya untuk maintenance dan web development sangat terjangkau.

Bahasa pemrograman PHP membantu Anda untuk mengembangkan aplikasi berbasis web yang cukup kompleks, handal, dan cepat. Tergantung dari spesifikasi bisnis, penggunaan hosting, tingkat pengalaman, kebutuhan aplikasi, dan pengembangan timeframe. Selain itu ada banyak PHP frameworks yang dapat Anda pilih.

Di bawah ini adalah contoh penggunaan PHP untuk menghubungkan dan membuat database.

```
<?php
$servername="localhost";
$username="username";
$password="password";
//Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database berhasil dibuat!";
} else {
    echo "Error membuat database: " . $conn->error;
}
$conn->close();
?>
```

## Sejarah PHP (*Hypertext Preprocessor*)

Pertama kali PHP dikenal dengan kependekan dari (*Personal Home Page*) atau Situs Personal yang dibuat oleh Rasmus Lerdorf pada Tahun 1995. Saat pertama kali dibuat PHP masih bernama *Form Interpreted* (FI) yang berupa sekumpulan skrip untuk pengolahan data formulir dari web.



Gambar 1. 1 Logo Pertama PHP

Kemudian Rasmus merilis kode sumber tersebut pada bulan November tahun 1997 untuk umum dengan memberikan nama PHP/FI. Dengan merilis kode tersebut menjadi kode sumber terbka sehingga banyak pemrograman yang tertarik untuk mengembangkan PHP tersebut.

Di tahun yang sama, sebuah perusahaan bernama Zend menulis kembali interpreter PHP menjadi lebih cepat, rapih dan lebih baik. Sehingga pada bulan Juni tahun 1998, perusahaan tersebut merilis kembali interpreter baru untuk PHP dan meresmikan perilisan tersebut dengan nama PHP 3.0 serta mengubah nama menjadi akronim berulang yang diberi nama PHP (*Hypertext Preprocessing*). Pada pertengahan tahun 1999, perusahaan Zend merilis kembali interpreter PHP dengan nama PHP 4.0 pada versi 4.0 ini banyak digunakan pada awal abad ke-21. Karena versi ini memiliki kemampuan untuk

membangun aplikasi web kompleks tetapi tetap memiliki stabilitas dan kecepatan yang tinggi. Setelah 5 tahun berlalu pada akhirnya Zend merilis kembali PHP dengan versi 5.0 pada bulan Juni tahun 2004. Pada versi ini, PHP memasukkan model pemrograman berorientasi objek untuk mengembangkan bahasa pemograman kea rah paradigm berorientasi objek.



Gambar 1. 2 Logo PHP 5

Di tanggal 18 Desember pada tahun 2014 Zend merilis bahasa pemrograman PHP menjadi versi 5.6.4 yang diperuntukan agar mempermudah pengembang menjalankan kode PHP tanpa menginstall *software server* terlebih dahulu. Tidak lama kemudian pada tanggal 17 Februari 2017 versi 5.6.4 di kembangkan kembali menjadi versi terbaru dan stabil pada bahasa pemrograman PHP saat ini menjadi versi 7.0.16 dan 7.1.2.



Gambar 1. 3 Logo PHP 7

### **Perkembangan PHP (*Hypertext Preprocessor*)**

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrograman yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni

1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Versi terbaru dari bahasa pemograman PHP adalah versi 5.6.4 yang resmi dirilis pada tanggal 18 Desember 2014 yang kemudian di perbarui lagi dengan versi teranyar yaitu PHP 7 yang di 17 Desember 2015.

### **Fungsi PHP (*Hypertext Preprocessor*)**

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap.

Sebagai perbandingan, website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari *user*, memproses form, dll.

Untuk pembuatan web, kode **PHP** biasanya di sisipkan kedalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai *Scripting Language* atau bahasa pemrograman *script*.

Sebagai contoh penggunaan PHP, misalkan kita ingin membuat list dari nomor 1 sampai nomor 10. Dengan menggunakan HTML murni, kita bisa membuatnya secara manual seperti kode berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<title>Contoh list dengan HTML</title>
</head>
<body>
<h2>Daftar Absensi Mahasiswa</h2>
<ol>
<li>Nama Mahasiswa ke-1</li>
<li>Nama Mahasiswa ke-2</li>
<li>Nama Mahasiswa ke-3</li>
<li>Nama Mahasiswa ke-4</li>
<li>Nama Mahasiswa ke-5</li>
<li>Nama Mahasiswa ke-6</li>
<li>Nama Mahasiswa ke-7</li>
<li>Nama Mahasiswa ke-8</li>
<li>Nama Mahasiswa ke-9</li>
<li>Nama Mahasiswa ke-10</li>
</ol>
</body>
</html>
```

Pada umumnya, untuk membangun halaman web, PHP bukanlah menjadi bahasa pemrograman yang utama. Menggunakan HTML dan CSS sudah dapat menghasilkan halaman web statis, dimana kontennya bersifat tetap. Bahasa PHP merupakan bahasa pemrograman pelengkap yang digunakan untuk menghasilkan web yang dinamis dan interaktif. Berikut akan dijelaskan mengenai beberapa fungsi dari PHP :

**a. Mempersingkat Tatanan HTML dan CSS**

Untuk membangun sebuah halaman web yang dinamis, PHP dapat berfungsi untuk mempersingkat penggunaan tatanan HTML dan CSS. Sebagai contoh, Dalam suatu sistem karyawan memiliki jumlah baris 100. Jika menggunakan HTML dan CSS maka barisan tersebut menjadi sangat panjang. Sedangkan jika ditambah dengan menggunakan PHP, maka Anda bisa mengatur beberapa baris yang dibutuhkan atau ditampilkan.

**b. Input Data**

Dengan menggunakan bahasa pemrograman PHP, Anda dapat menginput data dan menyimpannya dalam sistem Database, seperti MySQL.

**c. Manajemen Cookie dan Session**

Dalam PHP, Cookie dan Session digunakan untuk menyimpan informasi pengguna. Fungsi session\_start() untuk memulai session dan cookie dengan fungsi setcookie(). Sebagai contoh proses cookie, menyimpan username dan password pengguna di browser sehingga tidak harus mengisinya ulang ketika membuka situs yang sama. Session contohnya menyimpan informasi login yang hanya berlaku dalam satu sesi saja.

**d. Kompress Teks**

Dalam PHP, Anda bisa mengkompres teks yang panjang menjadi lebih pendek dengan fungsi gzcompress() dan mengembalikannya dengan fungsi gzuncompress().

Itulah beberapa fungsi dari PHP. Demikianlah penjelasan mengenai pengertian PHP, sejarah perkembangan PHP dan fungsi PHP. Penulis berharap agar Anda mengetahui sejarah perkembangan PHP dan fungsinya. PHP yang memiliki fungsi utama untuk mendesain web dinamis dan interaktif.

### **1.1.2 Pengantar CodeIgniter**

#### **1.1.2.1 Definisi CodeIgniter**

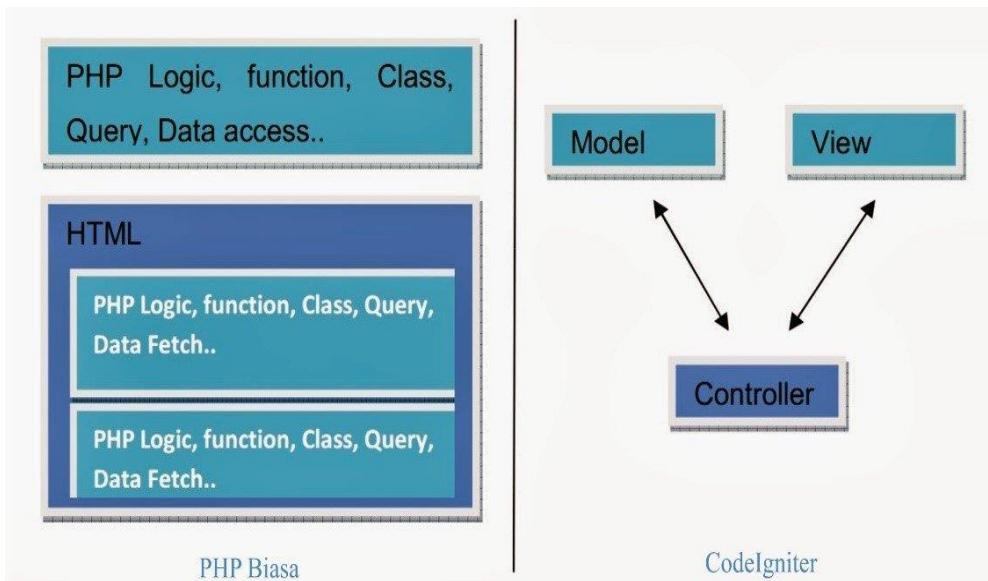
Framework atau dalam bahasa indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programer, tanpa harus membuat fungsi atau class dari awal.

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis.

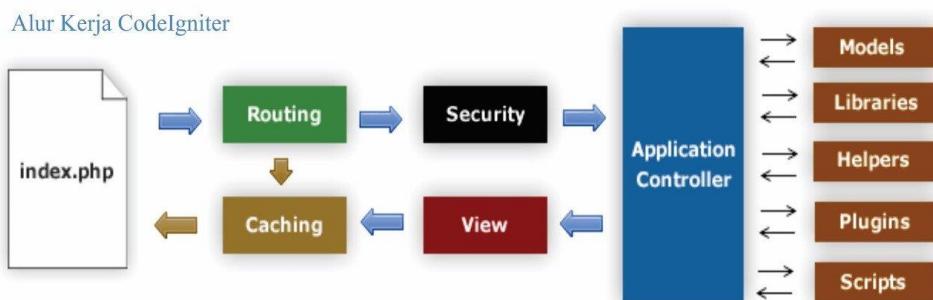
CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodennya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena “it is faster, lighter and the least like a framework.”

CodeIgniter pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, CodeIgniter dengan cepat “membakar”

semangat para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini.



Gambar 1. 4 Perbandingan PHP Biasa Dengan Codeigniter



Gambar 1. 5 Alur Kerja Framework Codeigniter

- **Index.php:** Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- **The Router:** Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.

- **Cache File:** Apabila dalam program sudah terdapat “cache file” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
- **Security:** Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter.
- **Controller:** Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut.
- **View:** Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di “cache” maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada.

```
login.php X

application > controllers > login.php

1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Login extends CI_Controller {
5      public function __construct()
6      {
7          parent::__construct();
8          $this->load->helper("url_helper");
9          $this->load->model('mlogin');
10         $this->load->library('session');
11     }
12     public function index_login()
13     {
14         $this->load->view("vlogin");
15     }
16     public function proses_login()
17     {
18         $data = array(
19             'username' => $this->input->post('username'),
20             'password' => $this->input->post('password')
21         );
22         // print_r ($data); die;
23         $hasil = $this->mlogin->cek_login($data);
24         if ($hasil->num_rows() == 1) {
25             foreach ($hasil->result() as $sess) {
26                 $sess_data['id_user'] = $sess->id_user;
27                 $sess_data['username'] = $sess->username;
28                 $sess_data['password'] = $sess->password;
29                 $sess_data['status_login'] = 'login';
30                 $sess_data['pw'] = $this->input->post('password');
31                 $this->session->set_userdata($sess_data);
32             }
33         }
34     }
35 }
```

Gambar 1. 6 Pembuatan Controller Pada Codeigniter

```
 surat.php X
application > models > surat.php
1  <?php
2  class surat extends CI_model{
3      public function __construct() {
4          $this ->load -> database();
5      }
6      public function get_masuk() {
7          return $this ->db ->get ('masuk') -> result_array();
8      }
9      public function deleteMasuk($id) {
10         $this ->db ->where ('no_surat', $id);
11         $this ->db ->delete('masuk');
12     }
13     public function masuk($no_surat) {
14         return $this->db->get_where('masuk', array('no_surat' => $no_surat)) ->row();
15     }
16     public function update_surat($id)
17     {
18         $data = array(
19             'no_surat' => $this->input->post('no_surat'),
20             'tgl_diterima' => $this->input->post('tgl_diterima'),
21             'sifat' => $this->input->post('sifat'),
22             'isi_ringkas' => $this->input->post('isi_ringkas'),
23             'dari' => $this->input->post('dari'),
24             'kepada' => $this->input->post('kepada'),
25             'diteruskan' => $this->input->post('diteruskan'),
26             'pengolah' => $this->input->post('pengolah'),
27             'file_surat' => $name
28         );
29         $this->db->where('no_surat',$id);
30         return $this->db->update('masuk', $data);
31     }
}
```

Gambar 1. 7 Pembuatan Model Pada Codeigniter

```
masuk.php ×
application > views > masuk.php
1  <!-- Content Wrapper. Contains page content --&gt;
2  &lt;div class="content-wrapper"&gt;
3      &lt;!-- Content Header (Page header) --&gt;
4      &lt;section class="content-header"&gt;
5          &lt;/section&gt;
6
7      &lt;!-- Main content --&gt;
8      &lt;section class="content"&gt;
9          &lt;div class="row"&gt;
10         &lt;div class="col-xs-12"&gt;
11             &lt;div class="box"&gt;
12                 &lt;div class="box-header"&gt;
13                     &lt;h3 class="box-title"&gt;Surat Masuk&lt;/h3&gt;
14                     &lt;h4 class="title"&gt;&lt;a href='&lt;?= base_url();?&gt;index.php/welcome/tambahmasuk/' class="btn btn-success"&gt;Tambah Masuk&lt;/a&gt;&lt;/h4&gt;
15                     &lt;div class="box-tools"&gt;
16                         &lt;div class="input-group input-group-sm hidden-xs" style="width: 150px;"&gt;
17                         &lt;/div&gt;
18                     &lt;/div&gt;
19                 &lt;/div&gt;
20                 &lt;div class="box-body table-responsive no-padding"&gt;
21                     &lt;div class="box"&gt;
22                         &lt;!-- /.box-header --&gt;
23                         &lt;div class="box-body"&gt;
24                             &lt;table id="example1" class="table table-bordered table-striped"&gt;
25                                 &lt;thead&gt;
26                                     &lt;tr&gt;
27                                         &lt;th width="50"&gt;&lt;center&gt;&lt;b&gt;No&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;
28                                         &lt;th width="10"&gt;&lt;center&gt;&lt;b&gt;No Surat&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;
29                                         &lt;th width="20"&gt;&lt;center&gt;&lt;b&gt;Tanggal Diterima&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;
30                                         &lt;th width="20"&gt;&lt;center&gt;&lt;b&gt;Sifat&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;
31                                         &lt;th width="20"&gt;&lt;center&gt;&lt;b&gt;Hal&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;
32                                         &lt;th width="10"&gt;&lt;center&gt;&lt;b&gt;Surat Dari&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;
33                                         &lt;th width="10"&gt;&lt;center&gt;&lt;b&gt;Kepada&lt;/b&gt;&lt;/center&gt;&lt;/th&gt;</pre>
```

Gambar 1. 8 Pembuatan View Pada Codeigniter

### 1.1.2.2 Sejarah CodeIgniter

*CodeIgniter* adalah aplikasi *open source* yang berupa *framework* dengan model **MVC (Model, View, Controller)** untuk membangun website dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *CodeIgniter* dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir 2.1.0 yang dirilis pada 28 November 2011.



*Gambar 1. 9 Logo CodeIgniter*

### **1.1.2.3 Perkembangan Machine Learning**

CodeIgniter selama ini masih dikembangkan walaupun relatif lambat, wacana perombakan dan inisiasinya dimulai pada bulan mei tahun 2015 lalu. Teman teman BCIT dan komunitas merencanakan untuk merombak core dari CodeIgniter supaya compatible dan menggunakan fitur PHP7. Beberapa hal yang dirombak meliputi Packages/Modules, Routing, Logging, Testing, dll.

Pengembangan tahap kedua fokus kepada pembuatan dan perombakan *class* yang sudah ada meliputi pembahasan berikut ini :

- Helpers
- Language/Localization
- Caching
- Email
- Encryption
- Form Validation
- Image Library

- Pagination
- Uploader
- Sessions
- Database adapters (RDBs)

Sejak Ellislab, perusahaan pengembang framework CodeIgniter dan CMS ExpressionEngine itu mengumumkan untuk melepas kepengurusan pengembangan CodeIgniter pertengahan tahun 2013 lalu, banyak sekali pihak yang mengajukan diri untuk melanjutkan pengembangan framework PHP populer itu. CodeIgniter yang pernah dikenal sebagai framework PHP yang paling cepat dan mudah dipelajari itu memang berkembang lebih lambat dibanding framework PHP lainnya seperti Symfony dan Yii. Ellislab memutuskan untuk melepas kepengurusan CodeIgniter --setelah sebelumnya mencoba melempar pengembangan ke komunitas, dengan alasan bisnis model yang sudah tidak cocok dengan visi bisnis Ellislab.

Hingga akhirnya 6 Oktober lalu Ellislab mengumumkan kandidat terpilih diantara banyak kandidat yang mengajukan diri untuk melanjutkan pengembangan CodeIgniter. **British Columbia Institute of Technology** (BCIT) yang berada di Vancouver Kanada-lah yang dianggap mumpuni untuk mengemban tanggung jawab komunitas dalam melanjutkan hidup CodeIgniter. Sebagai institut teknologi utama di Kanada, mereka mempunyai 480 ribu mahasiswa dan mengajarkan CodeIgniter kepada lebih dari 150 mahasiswa setiap tahunnya. James Parry sebagai salahsatu instruktur di BCIT School of Computing and Academic Studies yang akan memimpin proyek tersebut. Dialah yang akan membangun komite kepengurusan dan menghimpun komunitas untuk bersama-sama mengembangkan CodeIgniter menjadi framework yang lebih modern.

Seperti kita ketahui CodeIgniter adalah framework PHP yang paling populer terutama di Indonesia. Berdasarkan Google Trends, kata kunci

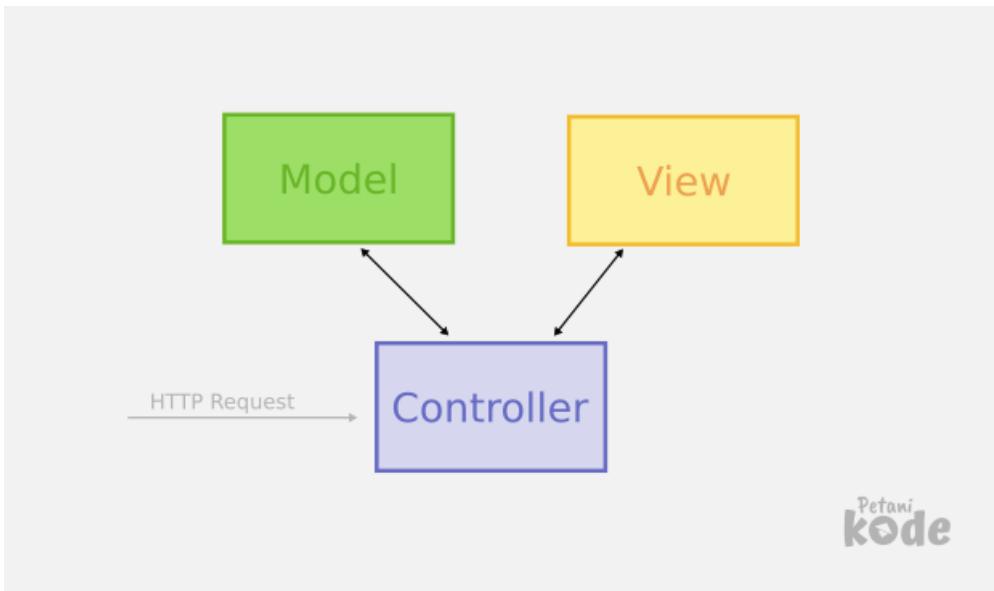
CodeIgniter menjadi yang tertinggi mengalahkan popularitas Zend Framework yang kian merosot sejak Mei 2013. Baru kemudian pada bulan Agustus 2014 popularitas Laravel --framework PHP yang muncul di pertengahan 2012 itu, mengalahkan CodeIgniter dengan kenaikan yang tajam. Hal tersebut kemungkinan disebabkan oleh lambatnya perkembangan CodeIgniter dalam mengikuti perkembangan PHP.

#### **1.1.2.4 Konsep MVC CodeIgniter**

MVC (*Model, View, Controller*) adalah sebuah pola desain (*design pattern*) arsitektur pengembangan aplikasi yang memisahkan dan mengelompokan beberapa kode sesuai dengan fungsinya.

MVC membagi aplikasi ke dalam tiga bagian fungsional: model, view, dan controller.

- a. **Model** adalah kode-kode untuk model bisnis dan data. biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
- b. **View** merupakan bagian yang menangani *presentation logic*. berisi kode-kode untuk tampilan.
- c. **Controller** merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



*Gambar 1. 10 Konsep MVC Codeigniter*

Alur kerjanya seperti ini:

1. Mulai;
2. User mengirim *request* ke web;
3. File yang pertama kali dieksekusi adalah index.php;
4. Lalu dari index.php, *request* akan diteruskan oleh routers.php;
5. routers.php akan mencari *cache* di server, apabila tedapat *cache* maka *cache* itu yang akan dikirim sebagai balasan (*response*). Apabila tidak ada *cache* barulah *request* diteruskan ke *Controller*;
6. *Controller* akan bertanggungjawab untuk mengambil data dari *Model* dan me-rendernya ke dalam *View* dengan menggunakan *library*, *plugin*, dan *helper* yang ada.
7. Hasil *render* (*view*) dikirim ke pengguna dan disimpan dalam *cache*, apabila fitur *cache* aktif;

## 8. Selesai.

### 1.1.3 Pengantar Database

Pengertian database – Di era yang modern ini tentu keberadaan teknologi menjadi hal yang biasa untuk ada di tengah masyarakat. Hal ini menjadi bukti dari semakin berkembangnya teknologi. Dan salah satu yang paling berkembang namun masih awam untuk diketahui adalah Database. Pada dasarnya database sudah akrab untuk digunakan sehari-hari, hanya saja kebanyakan dari kita masih belum mengerti pengertian dari teknologi yang satu ini.

Database sendiri pada dasarnya tidak hanya ditemui pada dunia IT saja, melainkan di berbagai bidang lainnya. Untuk itu agar mengetahui lebih lanjut seputar database, langsung saja simak ulasan berikut tentang pengertian database beserta fungsi dan jenisnya.

Seperti yang sudah dijelaskan sebelumnya bahwa database pada hakikatnya adalah suatu kumpulan data yang sebelumnya sudah dihimpun. Apabila dilihat dari segi bahasa, database juga bisa disebut dengan basis data. Sehingga dapat disimpulkan jika database merupakan kumpulan berbagai data dan informasi yang ada dan disimpan di suatu media tertentu, umumnya pada komputer.

Pengolahan database dalam komputer bertujuan untuk mempermudah akses komputerisasi. Pengolahan sistem database ini biasa juga disebut dengan istilah DBMS atau Database Management System yang juga bisa didefinisikan sebagai kumpulan tabel yang menyimpan data dan informasi.

Tetapi pada hakikatnya penerapan database ini termasuk dalam lingkup IT saja. Akan tetapi pemanfaatannya bisa dilakukan di berbagai bidang, seperti misalnya sekolah dan kampus yang wajib menampung seluruh data dari para staff dan siswanya.

### **1.1.3.1 Sejarah Database**

Tahun 1960

Dari awal penggunaan komputer, penyimpanan dan manipulasi data merupakan focus utama aplikasi. Pada awal tahun 1960, Charles Bachman diperusahaan General Electric mendesain generasi pertama DBMS yang disebut Penyimpanan Data Terintegrasi (Integrated Data Store). Dasar untuk model data jaringan dibentuk lalu distandardisasi oleh Conference on Data System Language (CODASYL). Kemudian, Bachman menerima ACM Turing Award (Penghargaan semacam nobel pada ilmu komputer) di tahun 1973.

Pada akhir tahun 1960-an, IBM mengembangkan system manajemen informasi (Information Manajemen System) DBMS. IMS dibentuk dari representasi data pada kerangka kerja yang disebut model data hierarki. Dalam waktu yang sama, hasil kerja sama antara IBM dengan perusahaan penerbangan Amerika mengembangkan system SABRE. System SABRE memungkinkan user mengakses data yang sama pada jaringan computer.

Tahun 1970

Pada tahun 1970, Edgar Codd di laboratorium penelitian di San Jose mengusulkan suatu representasi data baru yang disebut model data relational. Pada tahun 1980, model relasional menjadi paradigm DBMS paling dominan. Bahasa query SQL dikembangkan untuk basisdata relasional sebagai bagian proyek Sistem R dari IBM. SQL di standardisasi di akhir tahun 1980 dan SQL-92 diadopsi oleh American National Standards Institute (ANSI) dan International Standards Organization (ISO). Program yang digunakan untuk eksekusi bersamaan dalam basisdata disebut transaksi. User menulis programnya, dan bertanggung jawab menjalankan program secara bersamaan

terhadap DBMS. Pada tahun 1999, James Gray memenangkan Turing award untuk kontribusinya pada manajemen transaksi dalam DBMS.

## Tahun 1980

Pada akhir tahun 1980 dan permulaan tahun 1990, banyak bidang system basisdata dikembangkan. Penelitian dibidang basisdata meliputi bahasa query yang powerful, model data yang lengkap, dan penekanan pada dukungan analisis data yang kompleks semua bagian organisasi. Beberapa vendor (misalnya IBM, DB2, Oracle8, dan Informix UDS) memperluas sistemnya dengan kemampuan menyimpan tipe data baru misalnya image dan text serta kemampuan query yang kompleks. System khusus dikembangkan banyak vendor untuk membuat data warehouse dan mengonsolidasi data beberapa basisdata.

Suatu fenomena menarik adalah munculnya enterprise resource planning (ERP) dan management resource planning (MRP), yang menambah lapisan substansial dari fitur berorientasi aplikasi pada DBMS utama. Paket yang digunakan secara luas meliputi Baan, Oracle, PeopleSoft, SAP, dan Siebel. Paket tersebut mengidentifikasi kumpulan tugas umum (misalnya manajemen inventori, perencanaan sumber daya manusia, dan analisis keuangan) yang dihadapi oleh sejumlah besar organisasi dan menyediakan lapisan aplikasi umum untuk melaksanakan tugas. Data disimpan dalam DBMS relasional. Kemudian, lapisan aplikasi dapat disesuaikan pada perusahaan berbeda sehingga biaya keseluruhan perusahaan menjadi lebih rendah dibanding biaya pembuatan lapisan aplikasi dari awal. Lebih jauh, DBMS memasuki dunia internet. Saat generasi pertama, web site menyimpan datanya secara ekskulisif dalam file system operasi. Pada saat ini, DBMS dapat digunakan untuk menyimpan data yang dapat diakses melalui web browser. Query dapat dibuat melalui form web dan format jawabannya dengan menggunakan

markup language semisal HTML untuk mempermudah tampilan pada browser. Semua vendor basisdata menambah fitur ini untuk DBMS mereka.

Manajemen basisdata mempertimbangkan pentingnya suatu data bersifat online dan dapat diakses melalui jaringan computer. Saat ini, bidang seperti ini diwujudkan dalam basisdata multimedia, video interaktif, perpustakaan digital, proyek ilmuwan seperti proyek pemetaan, proyek sistem obeservasi bumi milik NASA, dan lain sebagainya (Ramakrishnan and Gehrke, 2003).

### **1.1.3.2 Struktur Database**

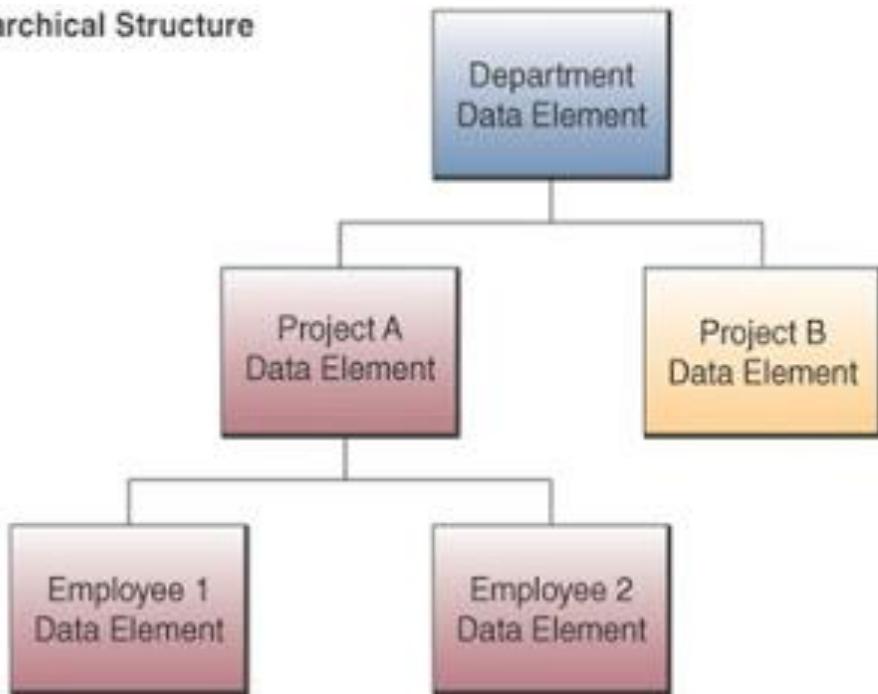
- 1 Hierarchical
- 2 Network
- 3 Relational
- 4 Object-oriented
- 5 Multi-dimensional

Hubungan antara berbagai elemen data terpisah yang disimpan dalam database didasarkan pada salah satu dari beberapa struktur atau model logis data. Software DBMS didesain untuk menggunakan struktur data tertentu agar dapat memberi para pemakai akhir akses yang cepat dan mudah ke informasi yang disimpan dalam database.

#### Hierarchical Structure

- Early DBMS structure
- Records arranged in tree-like structure
- Relationships are one-to-many

Hierarchical Structure

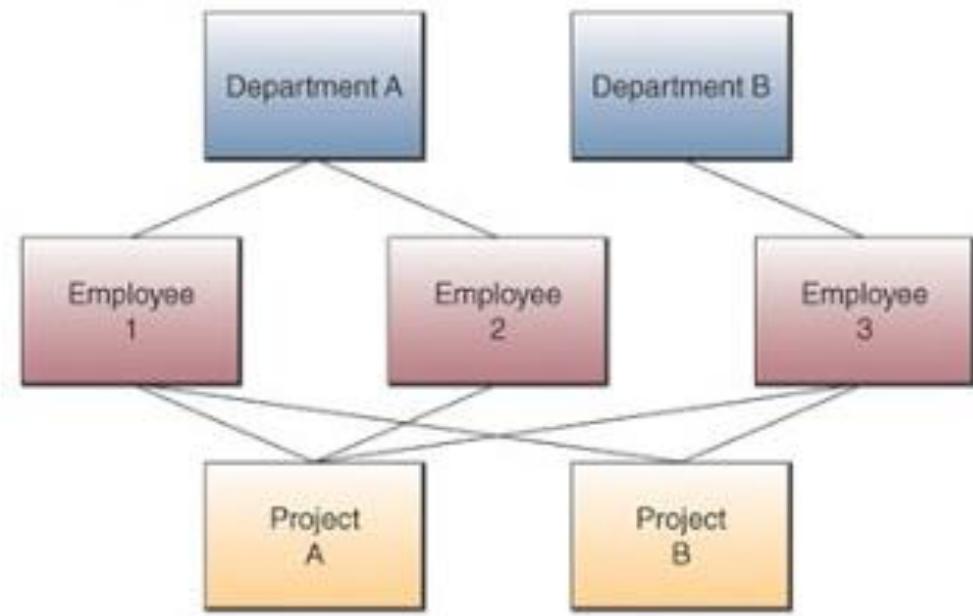


Gambar 1. 11 Struktur Hirarki Database

Software DBMS awal menggunakan struktur hirarkis, yang membuat hubungan antar catatan membentuk hierarki atau struktur yang seperti pohon. Dalam model hirarki tradisional, semua catatan merupakan dependen dan diatur dalam struktur multi tingkat, terdiri dari catatan akar (root) dan sejumlah tingkat subordinat. Semua hubungan antar catatan adalah satu-ke-banyak, karena setiap elemen data dihubungkan ke hanya satu elemen di atasnya.

#### Network Structure

- Used in some mainframe DBMS packages
- Many-to-many relationships



*Gambar 1. 12 Struktur Jaringan Database*

Struktur jaringan dapat mewakili hubungan logis yang lebih rumit, dan masih digunakan oleh beberapa software mainframe DBMS. Struktur ini memungkinkan hubungan banyak-ke-banyak antar catatan, dengan kata lain model jaringan dapat mengakses elemen data dengan mengikuti salah satu dari beberapa jalur, karena elemen data atau catatan apapun dapat dihubungkan ke banyak elemen data lainnya.

#### Relational Structure

The diagram illustrates a relational database structure with two tables: 'Department Table' and 'Employee Table'. Both tables are enclosed within a large oval boundary.

**Department Table**

Deptno	Dname	Dloc	Dmgr
Dept A			
Dept B			
Dept C			

**Employee Table**

Empno	Ename	Etitle	Esalary	Deptno
Emp 1				Dept A
Emp 2				Dept A
Emp 3				Dept B
Emp 4				Dept B
Emp 5				Dept C
Emp 6				Dept B

*Gambar 1. 13 Struktur Relasi Database*

Most widely used structure

- Data elements are stored in tables
- Row represents a record; column is a field
- Can relate data in one file with data in another, if both files share a common data element

Model relasional adalah model yang paling banyak digunakan dari ketiga struktur database. Model ini digunakan oleh kebanyakan software DBMS mikrokomputer, dan juga sistem skala menengah dan mainframe. Dalam model ini, semua elemen data dalam database disimpan dan dipandang dalam bentuk tabel-tabel sederhana.

### 1.1.3.3 Fungsi Database

Bagi masyarakat awam, mungkin database hanya diartikan sebagai kumpulan data dan informasi yang sepele. Padahal lebih daripada itu, database memiliki beberapa fungsi lain yang mampu mempermudah kinerja seseorang di era penyebaran informasi yang serba cepat ini. Beberapa fungsi lainnya diantaranya adalah sebagai berikut.

1. Sesuai dengan pengertian database yang mana memungkinkan kita untuk mengelompokkan data dengan tujuan untuk mempermudah identifikasi data. Pengelompokan bisa dilakukan melalui berbagai cara seperti misalnya membuat beberapa tabel atau *field* yang berbeda. Sebagai contohnya, hal ini sangat berperan bagi pihak perusahaan bank untuk mengumpulkan dan mengakses kembali data-data dari para nasabahnya.
2. Database juga mampu menghindari data yang tersimpan lebih dari sekali. Pada dasarnya semua kata perintah telah terekam di komputer sehingga bisa ketahuan apabila ada nama file ada data yang ganda.
3. Database juga mempermudah penggunaan di segala sisi, seperti saat memasukkan data baru ataupun mengupdate data dan menghapusnya. Tentunya database juga memungkinkan kita untuk melakukan riset dari jarak jauh karena sudah memiliki akses untuk menelpon dan berkomunikasi melalui nomor yang tertera.
4. Menggunakan teknologi ini juga memungkinkan kita untuk menghemat daya penyimpanan pada produk tertentu. Seperti misalnya kita bisa menyimpan data secara langsung melalui server dan jaringan internet sehingga tidak perlu lagi memberdayakan sumber daya manusia untuk menghimpun data yang jumlahnya sangat banyak itu.
5. Mampu Menghemat Kertas

Penggunaan database juga memungkinkan kita untuk menyimpan file secara digital sehingga tidak perlu lagi menggunakan kertas dalam jumlah banyak. Hal ini pastinya sangat praktis karena kita bisa langsung mengcopy setiap data yang diperlukan tanpa harus menulisnya kembali di kertas.

## **BAB II**

---

### **2.1 PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN**

#### **2.1.1 Tools Yang Digunakan**

##### **2.1.1.1 Visual Studio Code**

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst).



Gambar 2. 1 Logo Visual Studio Code

Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor-teks editor yang lain.

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files and folders, with 'views' selected. The Editor tab at the top has 'masuk.php - bla bla - Visual Studio Code'. The main Editor area displays the PHP code for 'masuk.php'. The Status Bar at the bottom shows 'Ln 32, Col 51 Spaces 2 UTF-8 CRLF No Notifications'.

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        </section>
    <!-- Main content -->
    <section class="content">
        <div class="row">
            <div class="col-xs-12">
                <div class="box">
                    <div class="box-header">
                        <h3>Surat Masuk</h3>
                        <h4><a href="#">Tambah Masuk</a></h4>
                        <div class="input-group input-group-sm hidden-xs" style="width: 150px;">
                            <input type="text" />
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="box-body table-responsive no-padding">
            <div class="box">
                <!-- /. box-header -->
                <div class="box-body">
                    <table id="example1" class="table table-bordered table-striped">
                        <thead>
                            <tr>
                                <th width="50"><center><b>No.</b></center></th>
                                <th width="10"><center><b>No Surat</b></center></th>
                                <th width="20"><center><b>Tanggal Diterima</b></center></th>
                                <th width="20"><center><b>Sifat</b></center></th>
                                <th width="20"><center><b>Hal</b></center></th>
                                <th width="10"><center><b>Surat Dari</b></center></th>
                                <th width="10"><center><b>Kepada</b></center></th>
                                <th width="10"><center><b>Diteruskan Kepada</b></center></th>
                                <th width="10"><center><b>Penialah</b></center></th>
                            </tr>
                        </thead>
```

Gambar 2. 2 Tampilan Visual Studio Code

VS Code dapat digunakan secara langsung tanpa perlu ekstensi selama alur pemrogramannya sudah didukung langsung. Namun ada beberapa alur pemrograman yang tidak didukung langsung misalnya penggunaan snippets code dan kemampuan Intellisense yang masih kurang. Disini letak kegunaan ekstensi karena dapat menambah kemampuan dukungan alur pemrograman yang diinginkan. Ekstensi VS Code ini banyak dibuat oleh pengembang-pengembang pihak ketiga yang juga menjadi kontributor dari VS Code. Ekstensi untuk VS Code dapat dilihat di Pasar Ekstensi dengan tautan berikut ini.

Daftar ekstensi tersebut akan dijelaskan secara singkat satu per satu berikut link keterangan untuk mengunduh ekstensinya melalui VS Code Marketplace.

1. **Angular Snippets by Dan Wahlin**. Ekstensi snippet dan suggestion untuk penggunaan framework Angular.
2. **Angular Snippets by Mikael Morlund**. Ekstensi snippet dan suggestion untuk penggunaan framework Angular.
3. **Angular Snippet by John Papa**. Ekstensi snippet dan suggestion penggunaan framework Angular.
4. **Angular Schematics**. Ekstensi menjalankan perintah Angular CLI dari File Explorer dan Command Palette VS Code.
5. **Angular Console**. Ekstensi untuk menjalankan Angular CLI dari VS Code dengan user interface atau GUI.
6. **Auto Close Tag**. Ekstensi untuk menambahkan penutup tag secara otomatis pada HTML dan XML . Ekstensi ini sementara tidak saya pakai karena bugs yang menyebabkan penggunaan CPU yang tinggi.

7. **Auto Rename Tag**. Ekstensi untuk mengganti tag pembuka dan tag penutup bersamaan pada HTML dan XML . Ekstensi ini sementara tidak saya pakai karena ada bugs yang menyebabkan penggunaan CPU yang tinggi.
8. **Autoprefixer** . Ekstensi untuk menambahkan prefix tipe browser secara otomatis dalam kelas CSS. Misalnya prefix -webkit-box untuk CSS Flexbox.
9. **Babel JavaScript** . Ekstensi untuk menampilkan highlight pada syntax JavaScript ES2015 ke atas.
10. **Beautify** . Untuk merapikan susunan kode JavaScript, HTML, CSS, dan JSON yang sesuai standar.
11. **Better Comment** . Ekstensi untuk memberikan tanda komentar yang lebih hidup dan mudah diperhatikan jika dibandingkan dengan komentar biasa di dalam kode.
12. **Bookmarks** . Menambahkan penanda bookmark pada baris kode dan navigasi ke penanda tersebut.
13. **Bracket Pair Colorizer** . Memberikan efek warna-warni pada pasangan bracket {}, (), dan [] di dalam kode.
14. **Bracket Pair Colorizer 2** . Versi terbaru dari ekstensi pewarnaan bracket sebelumnya. Versi terbaru ini membawa kompatibilitas dan performa lebih baik pada Visual Studio Code versi terbaru.
15. **Carbon-now-sh** . Ekstensi untuk membuat screenshot kode dengan layanan Carbon.

16. **CDNJS** . Mencari library dari CDNJS dan menambahkan langsung ke dalam HTML.
17. **Close HTML/XML Tag** . Memberikan tag penutup otomatis pada HTML dan XML.
18. **Color Info** . Ekstensi untuk menambahkan keterangan lebih lanjut dari warna di CSS.
19. **CSS Formatter** . Ekstensi untuk memformat susunan CSS agar lebih rapi.
20. **CSS Peek** . Melihat detail dari suatu kelas atau id dari CSS yang terdapat di dalam HTML .
21. **Debugger for Chrome** . Ekstensi penghubung antara VS Code dengan browser Google Chrome, yang digunakan untuk debugging aplikasi web.
22. **Dependency Analytics** . Melihat info dari dependency NPM yang dipakai.
23. **Docker** . Menambahkan syntax highlight dan linting untuk Dockerfile.
24. **DotENV** . Menambahkan syntax highlight dan linting untuk file .env .
25. **EditorConfig for VS Code** . Ekstensi untuk menyetel gaya penulisan di dalam panel teks editor, penggunaan spasi atau tab untuk mengatur jarak antar kode, besar jarak antar tab, jarak antar spasi, menambahkan satu baris di akhir baris kode, dst.
26. **ES7 React/Redux/GraphQL/React-Native snippets** . Ekstensi untuk menambahkan snippet dan syntax highlighting untuk React JS , Redux, dan GraphQL dengan ES2016.

27. **ESLint**. Ekstensi untuk mengatur gaya penulisan sesuai standar JavaScript ES2015+. Selain itu bisa memperbaiki kesalahan penulisan jika ditemui penulisan JavaScript yang belum sesuai standar.
28. **ES6 Mocha Snippets**. Ekstensi untuk membuat kerangka unit testing dengan Mocha.
29. **File Utils**. Ekstensi untuk menambahkan operasi membuat data, duplikasi data, memindahkan data, mengubah nama data, dan menghapus data dan direktori folder.
30. **Gi**. Membuat file .gitignore dengan mudah.
31. **Gitignore**. Dukungan syntax highlight dan mengambil kerangka gitignore dari repositori Gitignore di Github.
32. **Git History**. Ekstensi untuk melihat grafik riwayat commit dari repository Git di dalam folder proyek yang sedang dibuka. Selain itu bisa dilakukan operasi-operasi Git seperti branch, cherry pick, merge, dst.
33. **GitLens**. Ekstensi untuk meningkatkan kemampuan integrasi Git VCS di dalam Visual Studio Code.
34. **Git Graph**. Ekstensi menambahkan tampilan GUI dari alur Git dan status Git di VS Code langsung.
35. **Github Pull Request**. Membuat pull request dengan mudah di layanan Github.
36. **Highlight Matching Tag**. Menambahkan sorotan pada tag pembuka dan tag penutup pada elemen HTML.

37. **HTML CSS Support** . Menambahkan kemampuan Intellisense (suggestion dan completion) untuk penulisan class dan id dari CSS di dalam HTML.
38. **HTML Snippets** . Ekstensi untuk menambahkan dukungan terhadap penulisan HTML5. Namun fungsi HTML Snippets ini sudah diadopsi langsung oleh fungsi utama Visual Studio Code.\
39. **Indent Rainbow** . Ekstensi untuk menambahkan warna pada jarak tab dan spasi (indent) . Cukup membantu untuk menampilkan indent pembuka dan indent penutup. Terutama pada file HTML, JS, CSS, pada file Python, dst.
40. **IntelliSense for CSS class names in HTML** . Menambahkan kemampuan Intellisense terhadap pemakaian CSS Class di dalam HTML. Ekstensi ini akan melakukan pendekripsi untuk mencari file CSS yang terdapat di dalam folder proyek.
41. **Import Cost** . Melihat ukuran file yang diimport dari dependency dan file JS yang lain.
42. **Inline HTML** . Menambahkan syntax highlighting , snippet , dan intellisense pada sintaks HTML atau CSS yang berada di dalam ES2015 template Strings.
43. **JavaScript (ES6) code snippets** . Ekstensi untuk kerangka snippets penulisan JavaScript berdasarkan standar ES2015.
44. **JavaScript Snippet Pack** . Kumpulan snippet untuk JavaScript ES2015.
45. **Jest** . Menggunakan unit testing Jest dengan mudah.
46. **Jumpy** . Navigasi antar kata dengan pintasan cepat.

47. **JS Refactor** . Menambahkan fungsi refactor untuk kode JavaScript.
48. **Kubernetes** . Dukungan untuk melakukan debug aplikasi berbasis Kubernetes Container.
49. **Lit HTML** . Syntax highlighting dan Intellisense untuk sintaks HTML yang berada di dalam kode template String JavaScript.
50. **Live Server** . Ekstensi untuk menjalankan server lokal untuk proses debugging tampilan dan desain HTML. Live server ini akan mendeteksi perubahan di dalam file HTML, CSS, dan JS dan otomatis melakukan live reload halaman.
51. **Live Share** . Ekstensi untuk kolaborasi dengan pengembang lain secara online.
52. **Live Share Audio** . Ekstensi tambahan untuk Live Share dengan fungsi komunikasi melalui suara dengan pengguna Live Share yang lain.
53. **Markdown Preview Enhanced** . Ekstensi untuk menampilkan preview dari file Markdown.
54. **Markdown Lint** . Ekstensi untuk menambahkan snippet dan koreksi ketika penulisan Markdown.
55. **Markdown All In One** . Ekstensi untuk membantu penulisan markdown dengan fitur snippet, pintasan keyboard, preview, dan pembuatan daftar isi.
56. **MetaGo** . Ekstensi untuk membantu navigasi di dalam kode dengan teknik perpindahan antar karakter huruf kode yang sama.

57. **Multiple Clipboard** . Menambahkan dukungan untuk menampung banyak clipboard copy paste di VS Code.
58. **Node.js Modules Intellisense** . Ekstensi untuk menambahkan Intellisense terhadap daftar dependency di dalam Node Modules dan folder proyek. Sehingga ketika melakukan import statement di dalam kode JavaScript, terdapat saran atau suggestion dari nama module dependency nya.
59. **Node Readme** . Ekstensi untuk melihat dokumentasi dari library JavaScript yang telah terinstall di Node Module .
60. **NPM Support for VS Code** . Menambahkan fungsi untuk menjalankan script NPM melalui VS Code, inspeksi dependency yang ada di package.json , dan manajemen script NPM yang berjalan.
61. **NPM Intellisense** . Ekstensi ini hampir sama dengan ekstensi untuk NPM sebelumnya. Namun dilengkapi dengan perintah atau snippet yang bisa dipakai melalui Command Palette di VS Code.
62. **Open in Browser** . Ekstensi untuk membuka file HTML yang sedang disunting di VS Code ke browser komputer. Berguna untuk melihat tampilan web ketika proses desain dan mockup.
63. **Paste JSON as Code** . Melakukan copy paste data JSON menjadi bentuk model atau class.
64. **Path Intellisense** . Menambahkan Intellisense atau autocomplete pada penggunaan path di dalam HTML. Biasanya dipakai mereferensikan path dari file CSS ke dalam HTML.

65. **Path Autocomplete** . Fungsinya hampir sama dengan ekstensi Path Intellisense, namun ekstensi ini untuk mereferensikan file atau modul JavaScript yang satu dengan yang lain, ketika melakukan proses impor.
66. **Prettier Code Formatter** . Ekstensi untuk merapikan kode JavaScript, TypeScript, dan CSS.
67. **Polacode** . Ekstensi untuk mengambil tampilan atau screenshot dari beberapa potong kode. Berguna untuk membagikan potongan kode ke sosial media atau aplikasi chat.
68. **Polymer Snippet , Polymer Syntax , dan Polymer IDE** . Ekstensi untuk menambahkan fungsi snippet dan syntax highlighting pada framework Polymer dan Web Component
69. **Quokka.js** . Ekstensi untuk membuat lembaran coret-coret kode JavaScript, sebelum diimplementasikan ke proyek utama.
70. **React Native Tools** . Ekstensi untuk fungsi debugging pada framework React Native.
71. **React Native/ React / Redux Snippets for ES6 / ES7** . Ekstensi untuk menambahkan fungsi snippet pada pengembangan aplikasi React Native.
72. **React JS Code Snippets** . Ekstensi yang berisi snippet untuk React JS development dan sintaks ES2015.
73. **Regex Previewer** . Menambahkan preview pada fungsi Regex di JavaScript dan TypeScript.
74. **Remote Containers** . Ekstensi untuk membuka container Docker dari VS Code.

75. **Remote SSH** . Ekstensi untuk membuka folder dan kode dari remote server dengan SSH.
76. **REST Client** . Melakukan uji coba endpoint REST API dari Visual Studio Code langsung.
77. **Search node\_modules** . Menambahkan kemampuan untuk mencari file yang terdapat di dalam dependency pada folder node\_modules, melalui Command Palette VS Code.
78. **Settings Sync** . Ekstensi untuk melakukan backup data setelan VS Code ke dalam Github Gist. Data setelan yang dibackup diantaranya daftar ekstensi yang terpasang, setelan VS Code, key binding, dst . Dengan adanya fitur backup setelan ini, kita dapat melakukan ekspor dan impor setelan VS Code, ke mesin komputer yang lain. Atau untuk berjaga-jaga jika kita berganti komputer dan ingin memasang kembali ekstensi dan setelan yang pernah dipasang sebelumnya.
79. **Surround** . Menambahkan rangkap if else for dst dari selector kursor.
80. **Svelte** . Ekstensi untuk dukungan framework Svelte JS.
81. **Svelte Intellisense** . Menambahkan fungsi intellisense pada proyek berbasis Svelte JS.
82. **Svelte 3 Snippets**. Menambahkan snippets untuk Svelte JS versi 3.
83. **SVG Viewer** . Melihat preview untuk gambar SVG.
84. **Team Chat for Slack , Discord , Live Share** . Menambahkan fungsi berkirim pesan instan pada ekstensi Live Share.

85. **Trailing Spaces** . Menambahkan sorotan pada spasi kosong yang ada diakhiran kode dan menghapus spasi kosong tersebut.
86. **TSLint** . Memberikan fungsi linter untuk koreksi dan perbaikan kode TypeScript.
87. **Todo Highlight** . Highlight TODOs, FIXMEs, keywords, dan annotations .
88. **Todo List** . Menampilkan daftarakction comments (TODO, FIXME, HACK etc) ke dalam daftar yang mudah dibaca.
89. **Todo+** . Manajemen todo lists dengan mudah.
90. **Turbo Console Log** . Menambahkan pintasan cepat untuk menulis Console Log di JavaScript.
91. **Vetur** . Ekstensi yang menambahkan kemampuan snippets dan intellisense pada penulisan aplikasi web berbasis Vue JS .
92. **Visual Studio IntelliCode** . Menambahkan kemampuan Intellisense dengan AI dan suggestion yang tepat.
93. **VS Code Faker** . Membuat data palsu untuk alamat, nama, lorem ipsum, dan kontak untuk uji coba.
94. **VS Live Share** . Ekstensi untuk berbagi layar dan workspace di VS Code dengan pengguna atau tim proyek yang lain. Sangat berguna untuk melakukan pemrograman atau coding bersama , dengan pengguna yang terpisah jarak .

95. **VS Live Share Extension Pack** . Tambahan untuk ekstensi VS Live Share dengan fitur chatting dan mengobrol melalui suara.
96. **Vue 2 Snippets** . Ekstensi yang menambahkan kemampuan snippets dan intellisense pada penulisan aplikasi web berbasis Vue JS .
97. **Vue VSCode Snippets** . Ekstensi yang menambahkan kemampuan snippets dan intellisense pada penulisan aplikasi web berbasis Vue JS .
98. **Waka Time** . Mencatat statistik aktivitas programming kita yang dilakukan di VS Code.
99. **YAML** . Menambahkan dukungan untuk sintaks YAML .

### 2.1.1.2 XAMPP

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

Sejarah mencatat, software XAMPP pertama kali dikembangkan oleh tim proyek bernama Apache Friends dan sampai saat ini sudah masuk dalam rilis versi **7.3.9** yang bisa didapatkan secara gratis dengan label GNU (General Public License).

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini:

#### **X = Cross Platform**

Merupakan kode penanda untuk software cross platform atau yang bisa berjalan di banyak sistem operasi.

## **A = Apache**

Apache adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (open source).

## **M = MySQL / MariaDB**

MySQL atau MariaDB merupakan aplikasi database server yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database.

## **P = PHP**

Huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat website dinamis, contohnya dalam website berbasis CMS WordPress.

## **P = Perl**

Sementara itu, untuk huruf P selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan.



*Gambar 2. 3 Logo XAMPP*

## **Bagian-bagian Penting di Dalam XAMPP**

### **1. Htdocs**

Htdocs merupakan nama sebuah folder bagian dari XAMPP yang berfungsi untuk menyimpan berbagai file dan dokumen yang akan ditampilkan kedalam website. Kapasitas penyimpanan di dalam menu htdocs sendiri bergantung kepada harddisk yang terpasang di perangkat komputer.

### **2. Control Panel**

Seperti namanya, Control Panel memberikan kita akses lebih leluasa dalam mengatur database, mengunggah file, atau melakukan setting lebih

detail terkait jeroan website. Dari dalam Control Panel kita juga bisa mengatur stop dan start aplikasi XAMPP agar tidak memberatkan kinerja komputer.

### **3. PhpMyAdmin**

Sama seperti fungsi PhpMyAdmin pada web hosting sesungguhnya, peran PhpMyAdmin dalam aplikasi XAMPP juga diplot sebagai pengatur konfigurasi MySQL. Untuk membuka phpMyAdmin di XAMPP caranya Anda bisa langsung mengetikkan URL <http://localhost/phpMyAdmin>.

Demikianlah artikel terkait pengertian XAMPP sebagai salah satu software pendukung untuk membantu proses belajar para pengembang web pemula. Jika Anda sudah merasa puas dan yakin dengan kinerja fitur maupun tampilan desain website Anda, jangan lupa untuk menggunakan paket hosting murah dari Qwords.com ya agar situs Anda bisa diakses oleh lebih banyak pengunjung tanpa perlu khawatir baakal menghabiskan banyak biaya.

#### **2.1.2 Bahasa Pemograman**

##### **PHP**

Bahasa pemrograman PHP biasanya tidak digunakan pada keseluruhan pengembangan website, melainkan dikombinasikan dengan beberapa bahasa pemrograman lain. Misalnya saja untuk mengatur tampilan, layout, dan berbagai macam menu menggunakan CSS.

Selain itu, terdapat juga beberapa framework PHP; Laravel, Phalcon, Codigniter, Symfoni yang saat ini banyak tersedia di internet untuk memudahkan proses pengembangan website menggunakan bahasa pemrograman tersebut.

PHP saat ini bisa dijalankan di berbagai platform system operasi, diantaranya adalah sebagai berikut :

- Linux/Unix
- Windows
- Mac OS X
- Java Virtual Machine
- OS/2
- Amiga
- Palm
- Symbian (untuk produk-produk Nokia)

Bahasa pemograman ini dihubungkan dengan beberapa lisensi yang berbeda dari beberapa versi. Lihat sejarahnya di Python Copyright. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial.

## BAB III

### 3.1 INSTALASI TOOLS YANG DIGUNAKAN

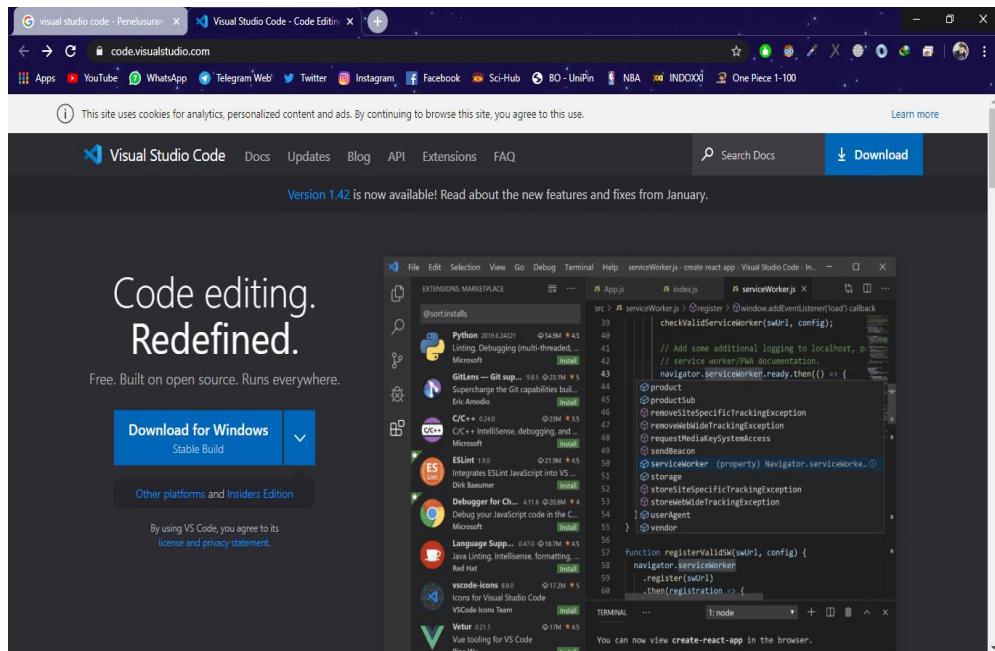
#### 3.1.1 Tools Yang Digunakan

##### 3.1.1.1 Instalasi Visual Studio Code

Berikut adalah langkah-langkah untuk melakukan instalasi Sublime :

1. Pertama kita harus download terlebih dahulu file intaller Visual Studio Code melalui situs resminya :

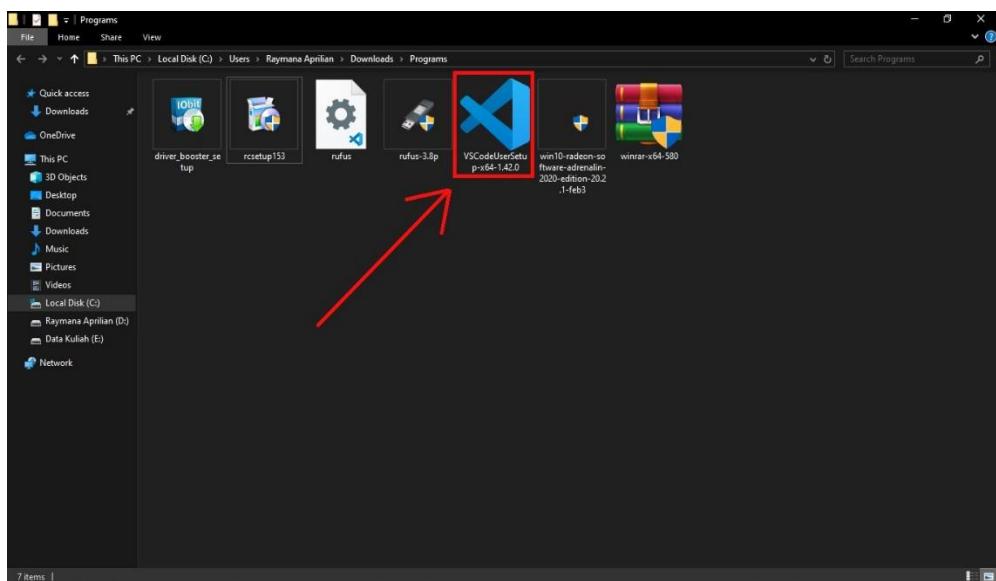
<https://code.visualstudio.com/>



Gambar 3. 1 Download Visual Studio Code

Jika sudah memiliki file .exe nya silahkan lanjutkan ke langkah instalasi berikutnya, tetapi jika belum memiliki file .exe nya silahkan lakukan download terlebih dahulu.

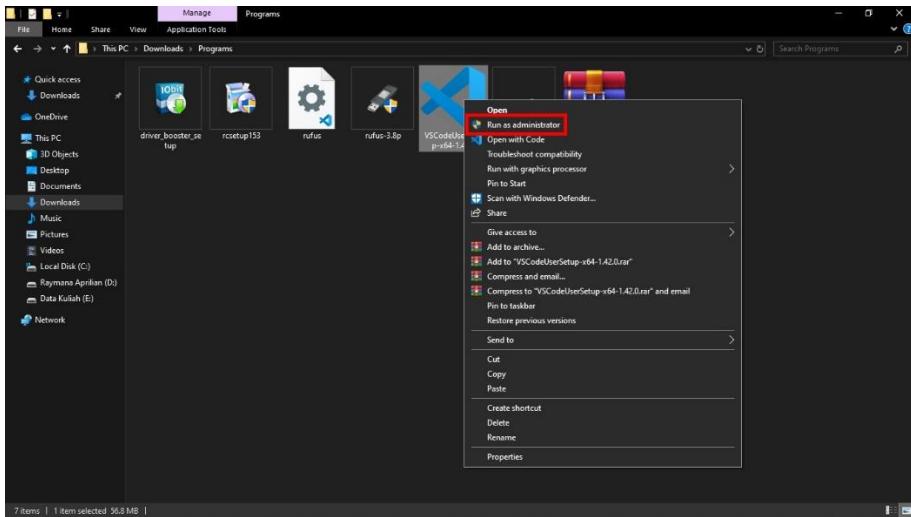
2. Double klik pada file installer nya atau klik kanan kemudian pilih Run as Administrator :



*Gambar 3. 2 File .exe*

Setelah selesai mengunduh file .exe atau aplikasi nya, cari file .exe nya pada direktori download, kemudian klik dua kali pada file .exe nya atau bias menggunakan klik kanan kemudian Run As Administrator.

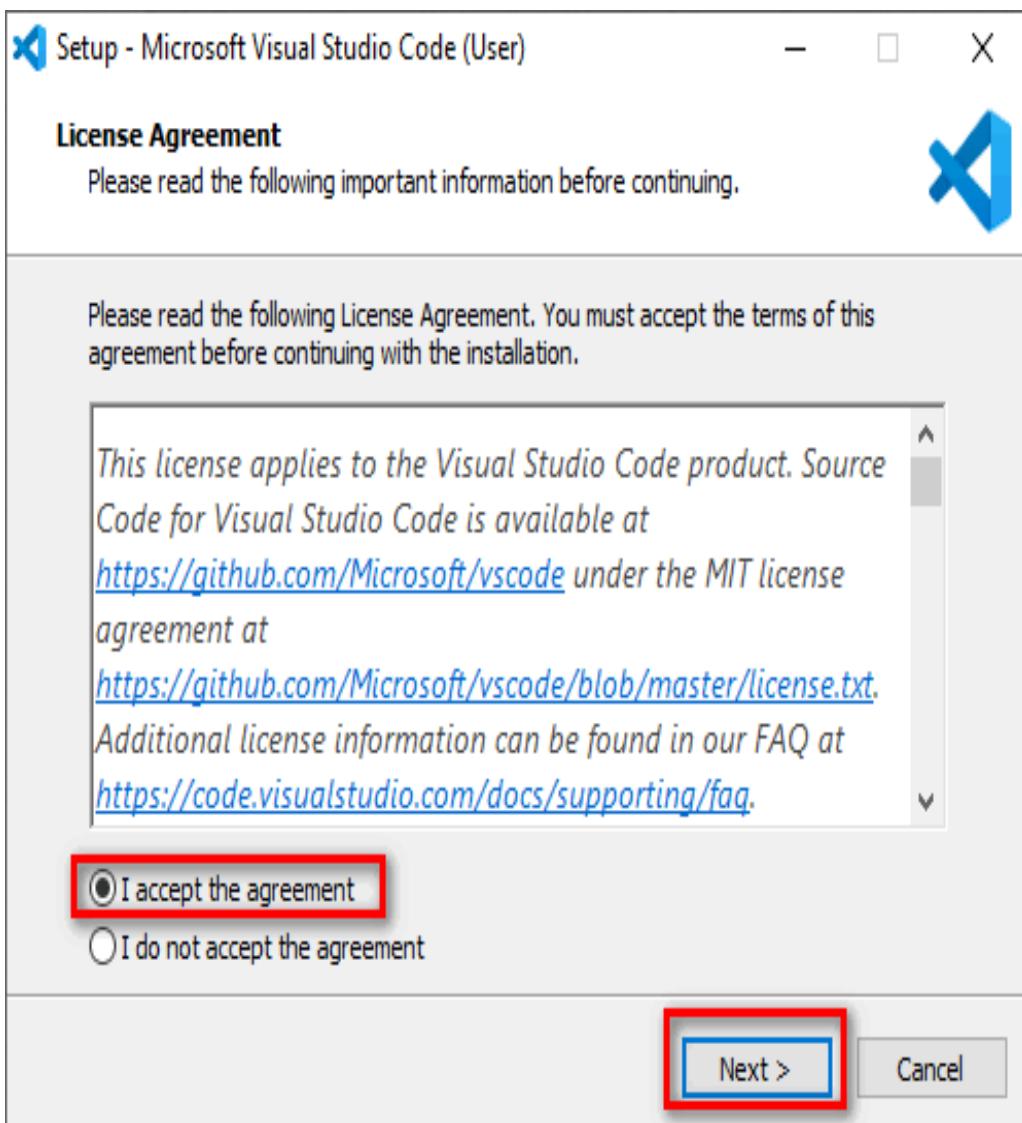
3. Jika muncul peringatan Run as Administrator, silahkan klik Yes.



*Gambar 3. 3 Pilih “Run As Administratosr”*

Selanjutnya klik yes dan masuk ke dalam proses instalasi berikutnya, dan ikuti langkah – langkah selanjutnya.

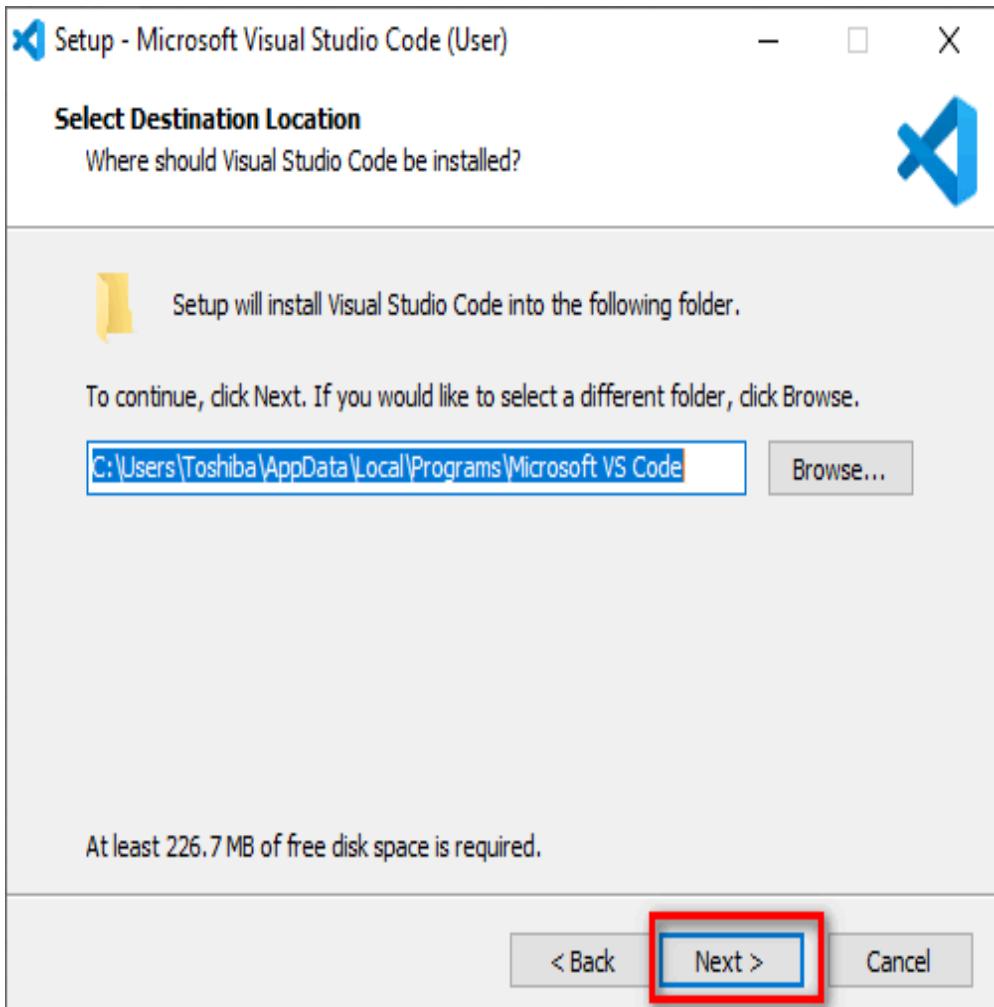
4. Pilih “I accept the aggrement” untuk menyetujui “License Agreement”, kemudian klik Next.



Gambar 3. 4 Pilih “I accept the agreement”

Dalam proses ini ikutilah sesuai gambar yang diberi tanda merah dan lanjutkan

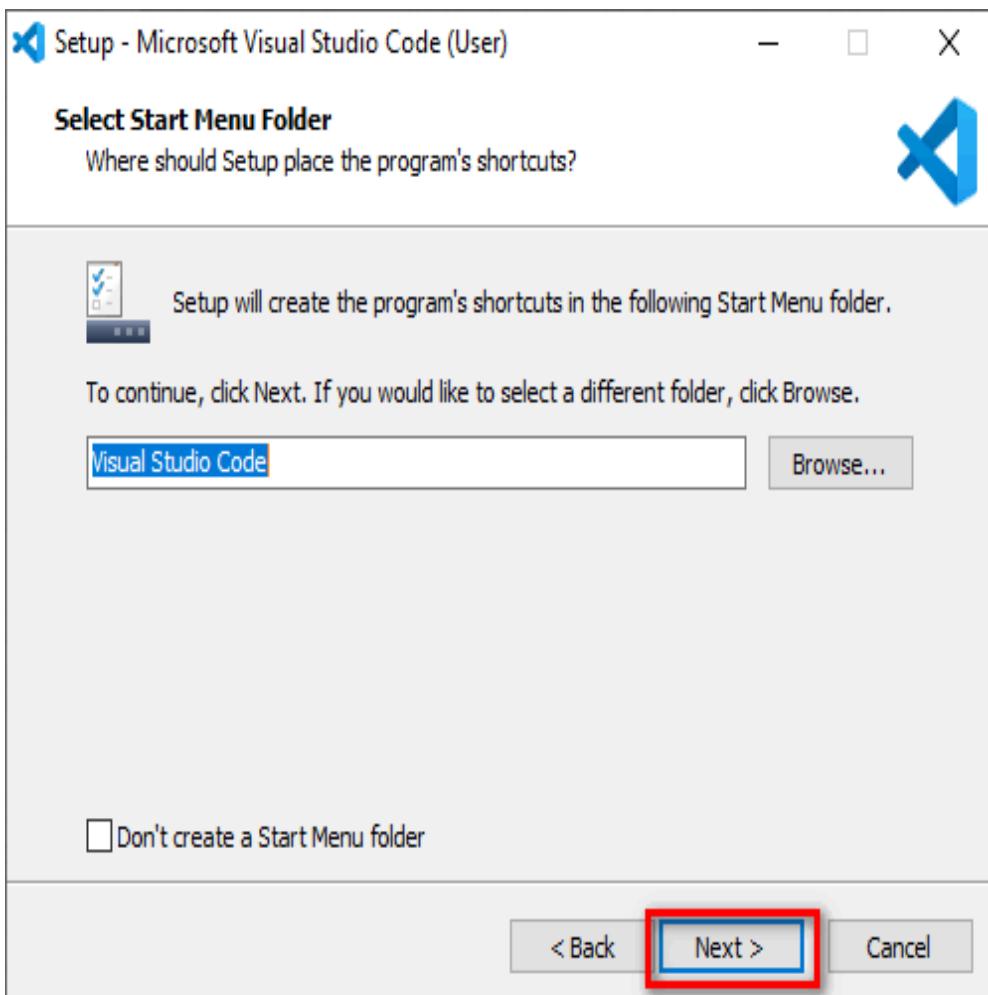
5. Untuk Select Destination Location bisa di biarkan saja jika lokasi installasi tidak akan di rubah. Klik Next.



*Gambar 3. 5 Memilih Tempat Penyimpanan*

Dalam proses ini kita bias menentukan sendiri tempat penyimpanan folder hasil instalasi nya, jika tidak ingin mengganti silahkan gunakan direktori yang sudah ditentukan atau default.

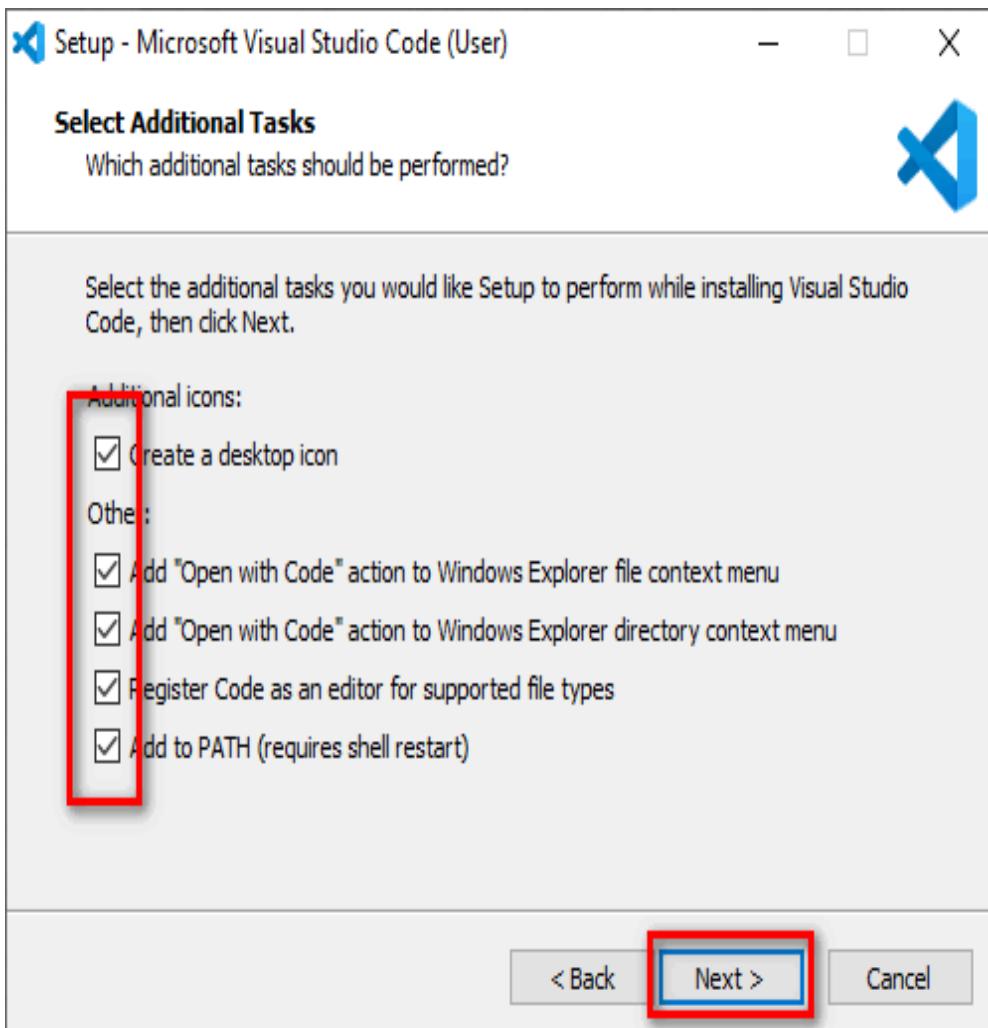
6. Klik Next lagi jika tidak akan merubah Start Menu Folder.



*Gambar 3. 6 Menentukan Nama Folder*

Disini kita juga bebas memberi nama folder yang akan menyimpan hasil instalasinya jika tidak ingin mengganti nama folder tersebut gunakan nama folder yang sesuai dengan nama aplikasinya atau yang sudah diberikan secara default.

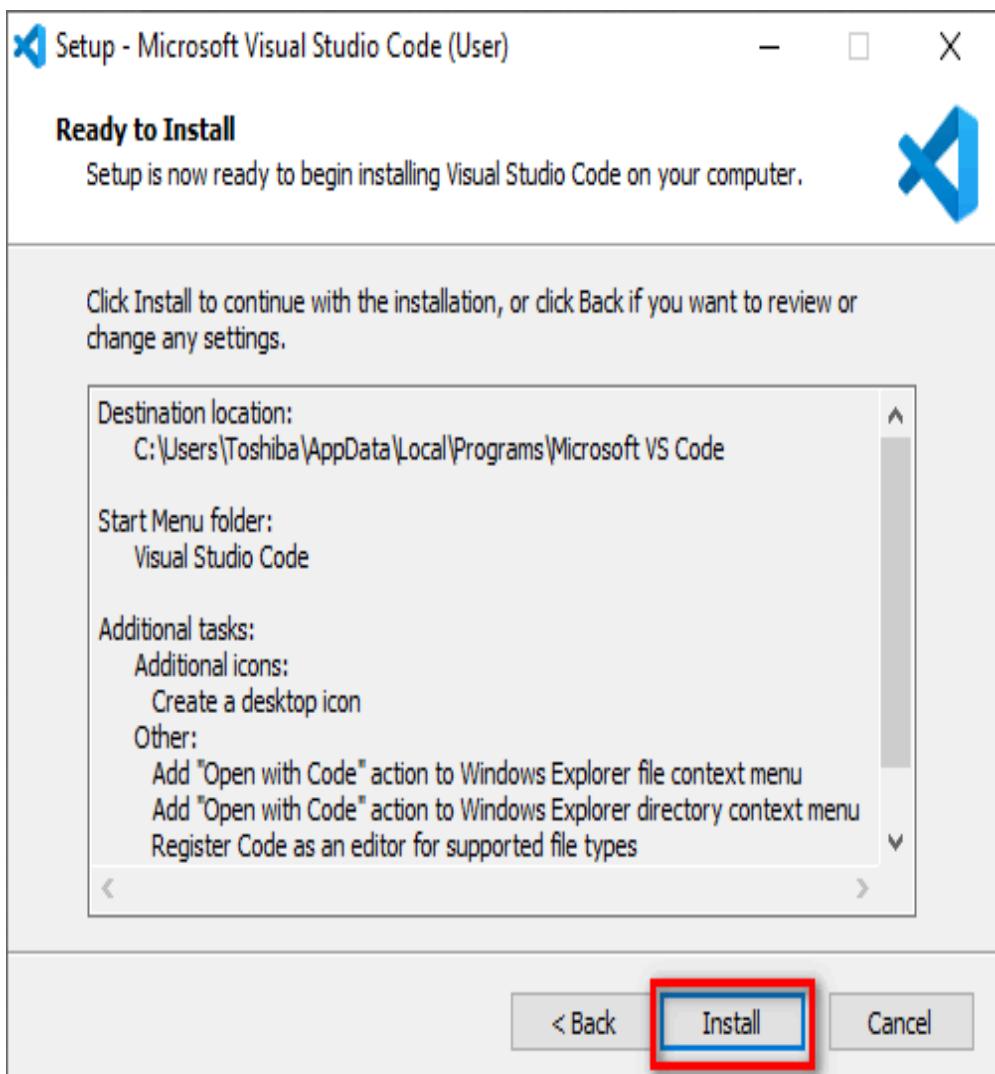
7. Di bagian Select Additional Tasks centang semua. Kemudian Next



Gambar 3. 7 Ceklis Aditonal Tasks

Dalam proses ini kita di anjurkan untuk mencontreng semua yang ada pada Aditonal icon dan other lalu lanjutkan dengan menekan tombol “Next”.

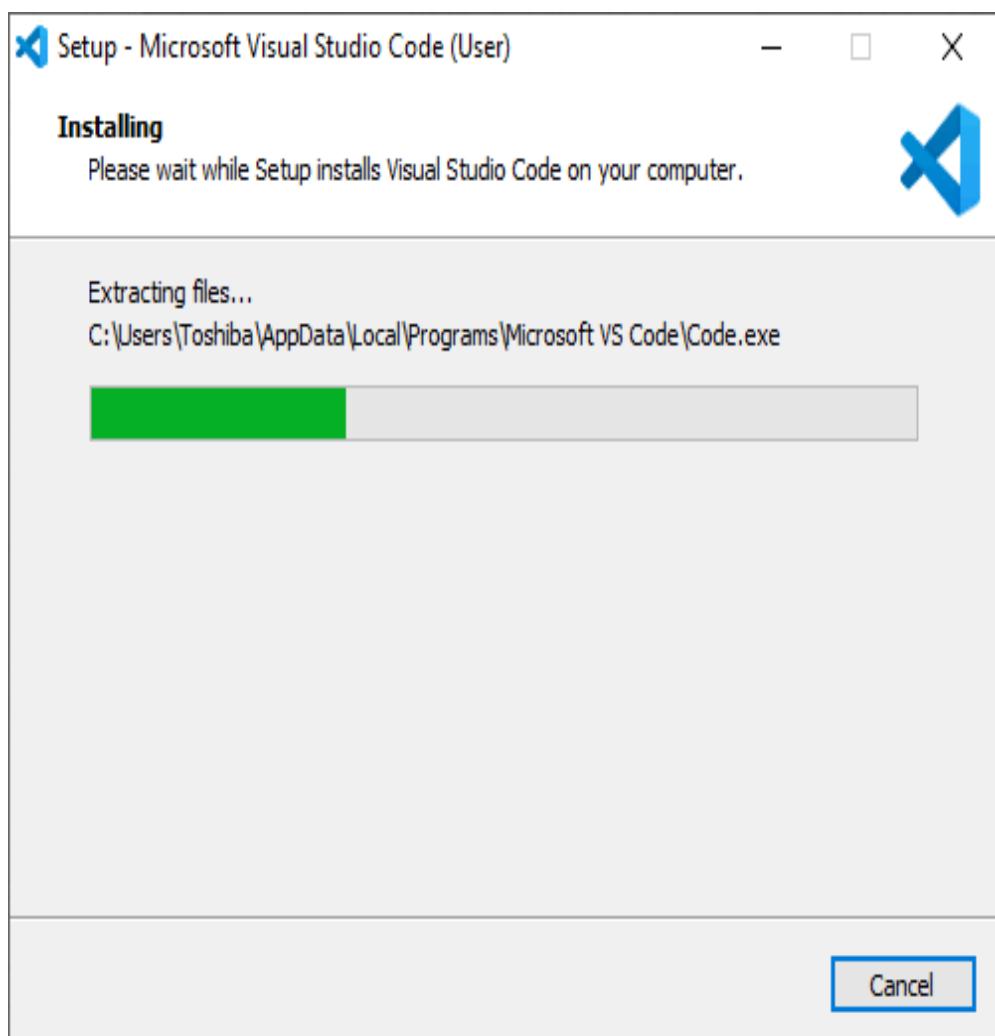
8. Lalu klik Install untuk memulai proses installasi.



Gambar 3. 8 Siap di Install

Disini pembaca hanya perlu menekan tombol “Install” saja untuk melanjutkan proses instalasi.

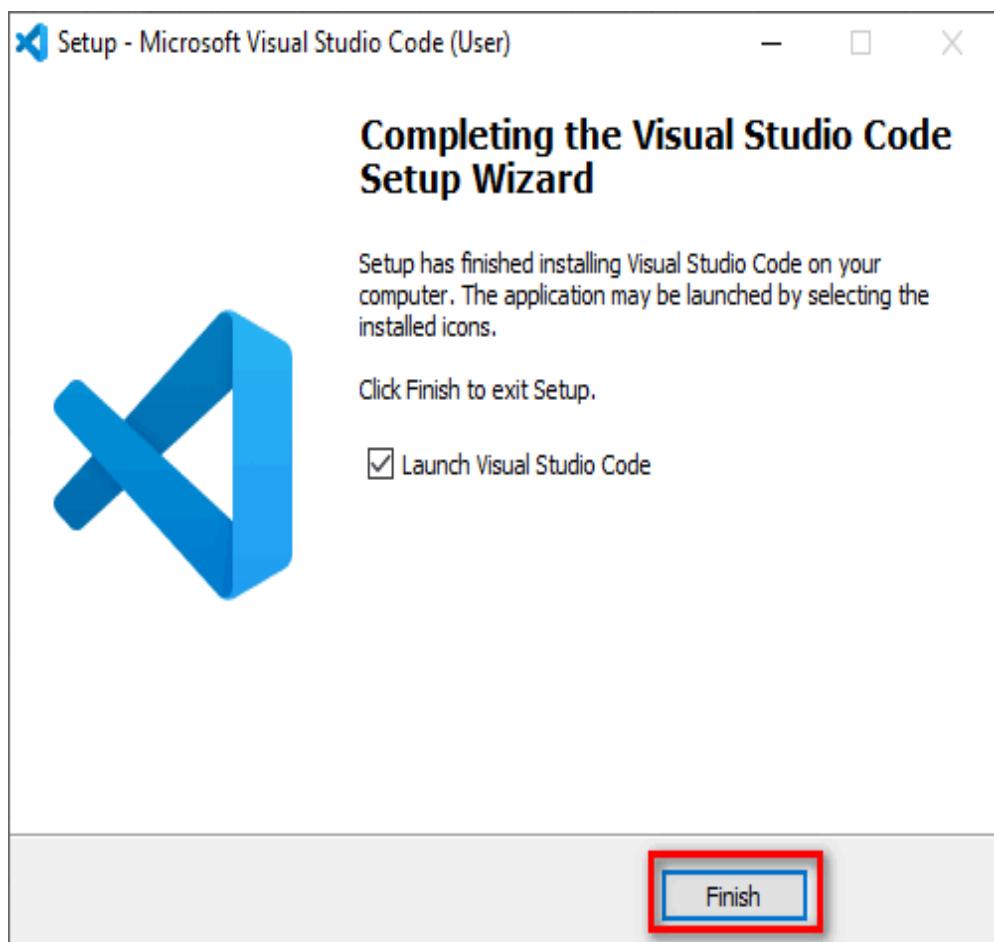
9. Tunggu sampai proses installasi selesai.



*Gambar 3. 9 Proses Instalasi*

Ini adalah proses instalasi dari aplikasi Visual Studio Code, tunggu hingga prosesnya selesai.

10. Setelah selesai klik “Finish”



Gambar 3. 10 Instalasi Selesai

Jika proses instalasi sudah selesai centang pada “Launch Visual Studio Code” dan tekan tombol “Finish”. Jika sudah aplikasi sudah dapat digunakan.

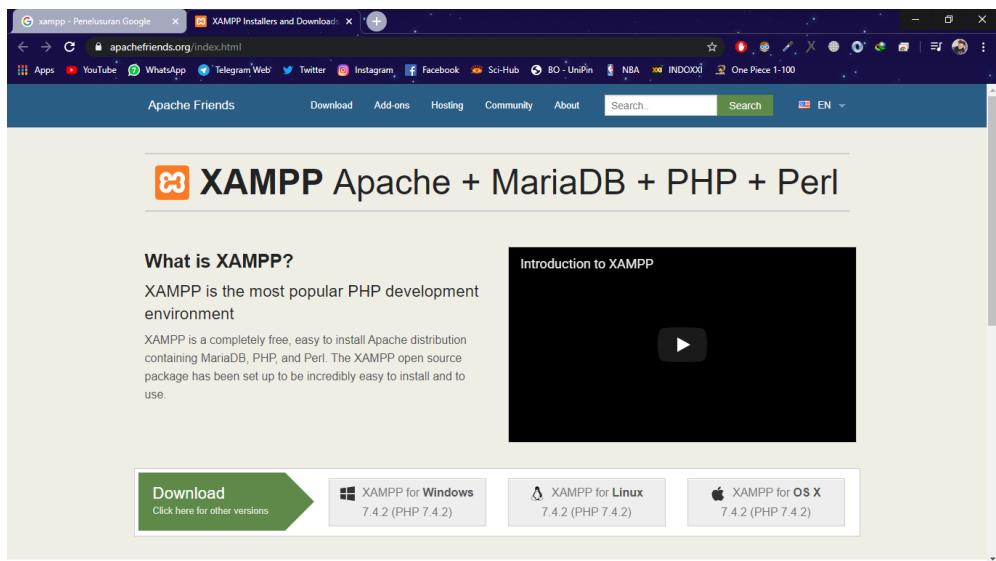
### 3.1.1.2 Instalasi XAMPP

Berikut adalah langkah-langkah untuk melakukan instalasi XAMPP :

1. Untuk melakukan instalasi xampp maka pertama-tama melakukan download file .exe pada link berikut ini :

<https://www.apachefriends.org/index.html>

Sesuai kan dengan system operasi yang anda gunakan.

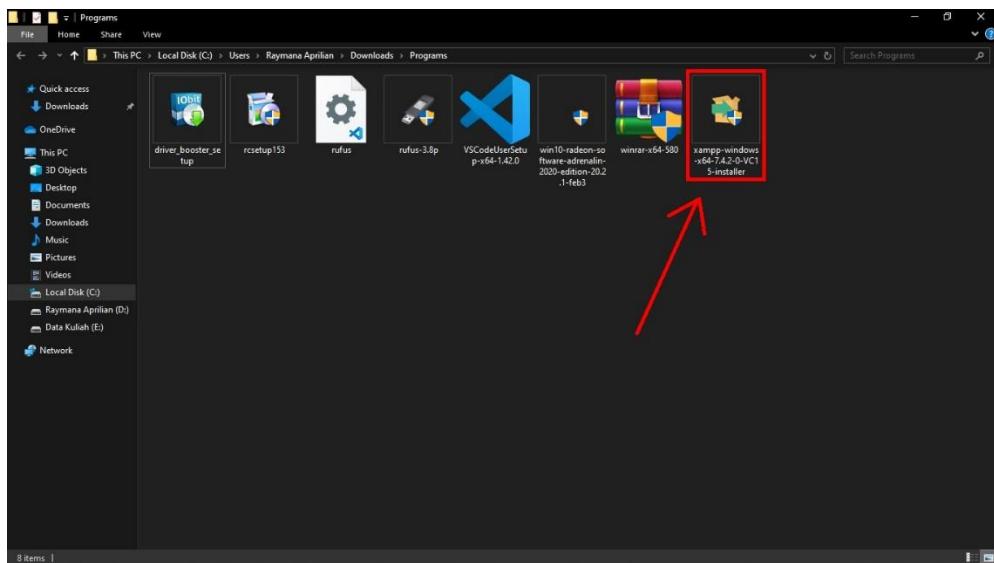


Gambar 3. 11 Download File .exe

Pada langkah pertama ini pembaca di anjurkan untuk mendownload file .exe xampp pada situs resmi apache friends, atau bias menyalin link yang

telah dituliskan di atas, jika sudah memiliki file.exe nya maka lewati langkah ini dan lanjutkan ke langkah berikutnya.

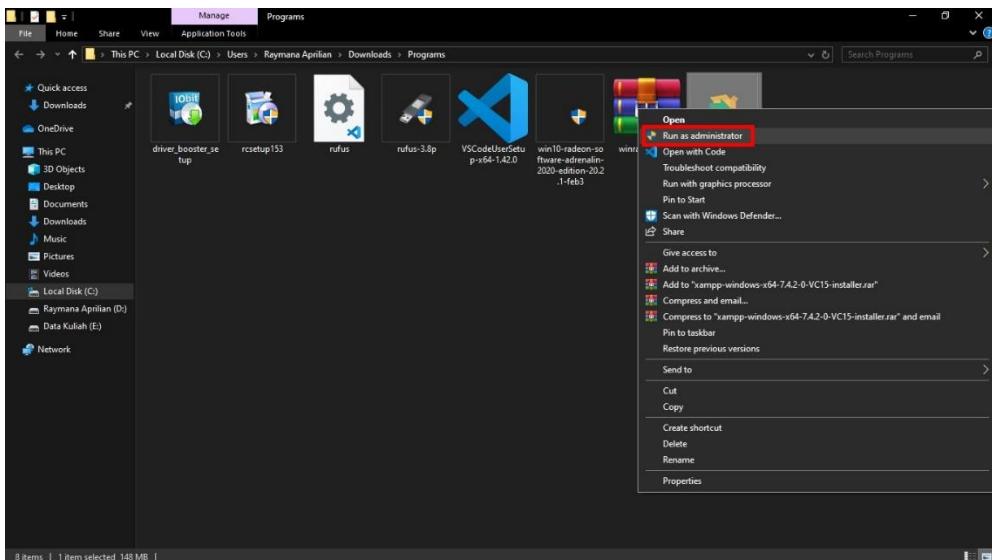
2. Double klik pada file installer nya atau klik kanan kemudian pilih Run as Administrator :



*Gambar 3. 12 Doble klik pada file .exe*

Setelah selesai mengunduh file .exe atau aplikasi nya, cari file .exe nya pada direktori download, kemudian klik dua kali pada file .exe nya atau bias menggunakan klik kanan kemudian Run As Administrator.

3. Jika muncul peringatan Run as Administrator, silahkan klik Yes.



Gambar 3. 13 Klik kanan pada file .exe lalu klik “Run As Administrator”

Selanjutnya klik yes dan masuk ke dalam proses instalasi berikutnya, dan ikuti langkah – langkah selanjutnya.

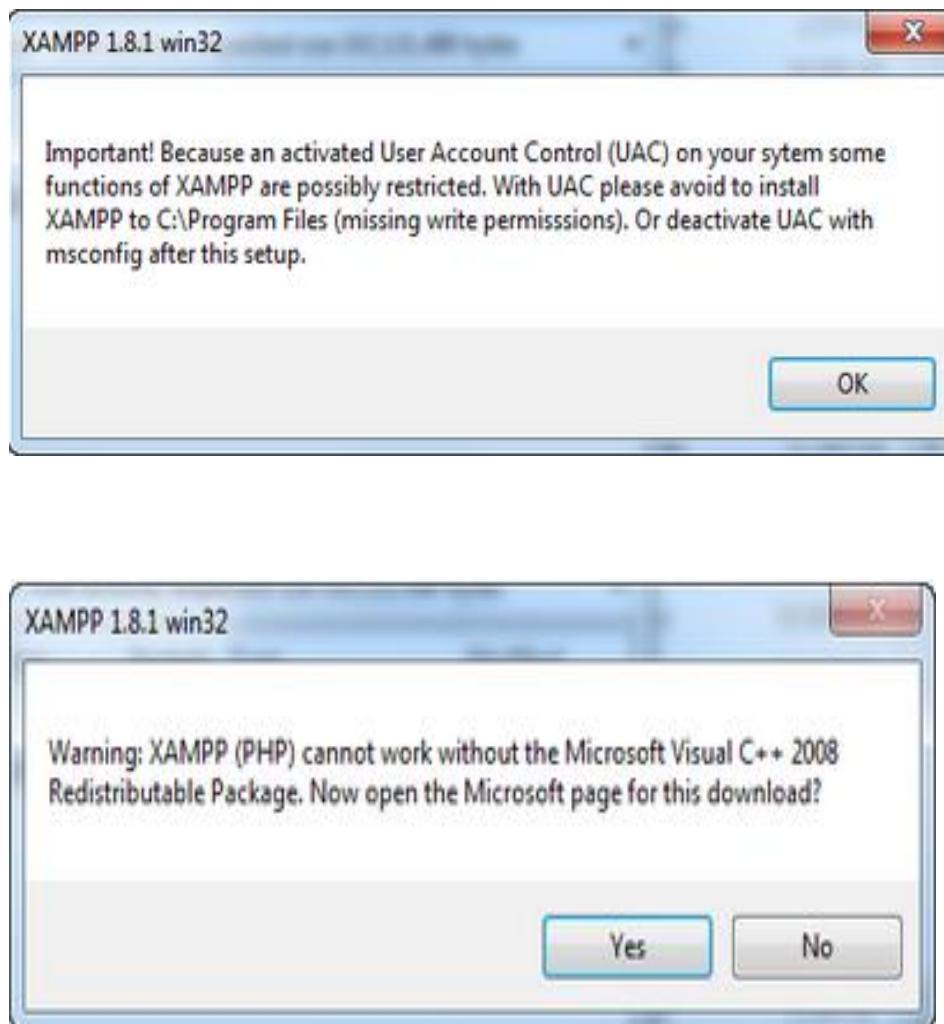
4. Selanjutnya pilih bahasa. Pilih yang Bahasa Inggris (English). Klik OK.



Gambar 3. 14 Bahasa Instalasi

Pada langkah ini pembaca bias memilih Bahasa yang akan digunakan dalam instalasi, penulis menyarankan untuk menggunakan Bahasa inggris, setalah memilih tekan “OK” dan akan lanjut ke langkah berikutnya.

5. Kadang pada proses ini muncul pesan error. Jika ada, abaikan saja dan lanjutkan dengan klik OK dan YES.



Gambar 3. 15 Error Package

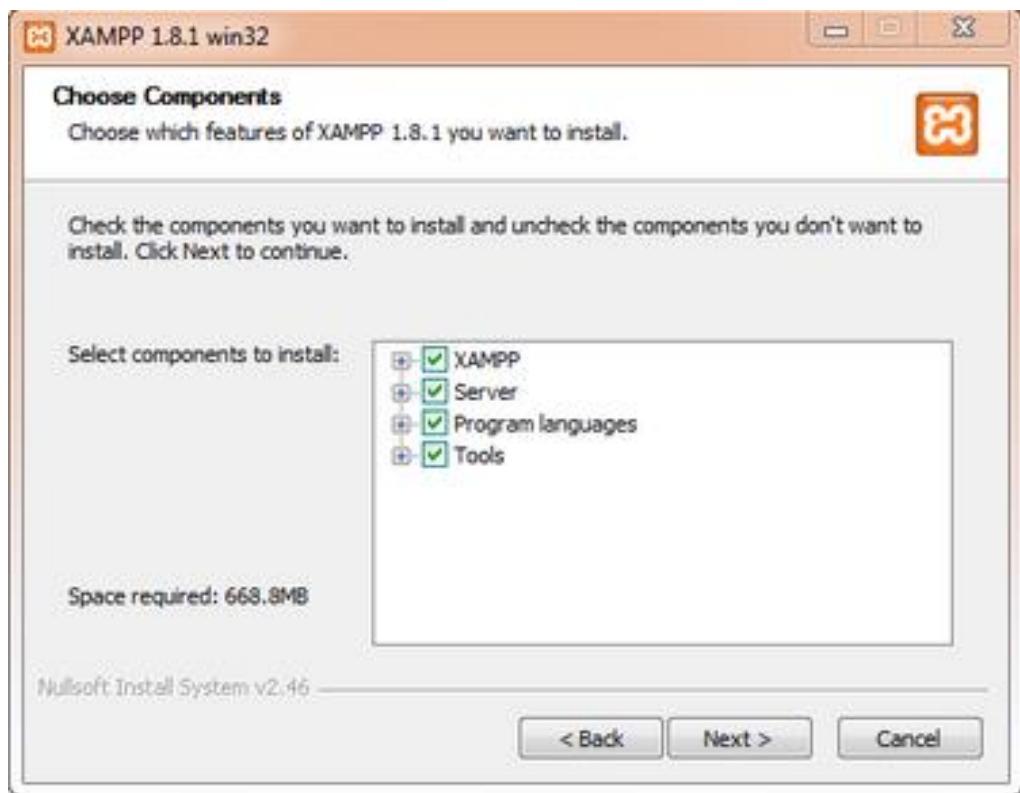
Jika muncul error seperti gambar di atas maka abaikan saja dan lanjutkan langkah instalasi.

6. Berikutnya akan muncul jendela yang isinya meminta Anda menutup semua aplikasi yang sedang berjalan. Jika semua aplikasi sudah ditutup, maka klik tombol “Next”.



*Gambar 3. 16 Tampilan Instalasi*

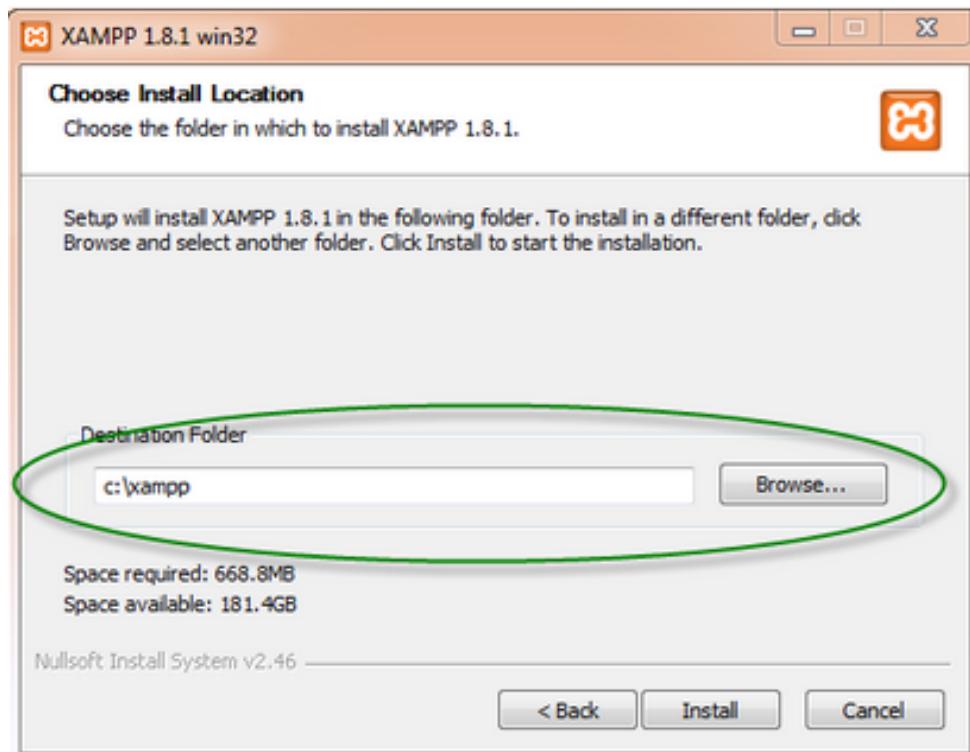
7. Selanjutnya Anda akan diminta untuk memilih aplikasi yang mau diinstal. Centang saja semua pilihan dan klik tombol **Next**.



*Gambar 3. 17 Memilih Komponen*

Pada langkah ini pembaca disarankan untuk memilih komponen dalam instalasi, penulis menyarankan untuk memilih semua komponen tersebut.

8. Kemudian Anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder XAMPP. Secara default akan diarahkan ke lokasi **c:\xampp**. Namun jika Anda ingin menyimpannya di folder lain bisa klik **browse** dan tentukan secara manual folder yang ingin digunakan. Jika sudah selesai, lanjutkan dan klik tombol **Install**.



*Gambar 3. 18 Memilih File Location*

Dalam proses ini kita bias menentukan sendiri tempat penyimpanan folder hasil instalasi nya, jika tidak ingin mengganti silahkan gunakan direktori yang sudah ditentukan atau default.

9. Tunggu beberapa menit hingga proses intalasi selesai. Jika sudah muncul jendela seperti di bawah ini, klik tombol **Finish** untuk menyelesaiakannya.



Gambar 3. 19 Instalasi Selesai

Proses instalasi xampp sudah selesai dan siap digunakan dalam pembuatan aplikasi.

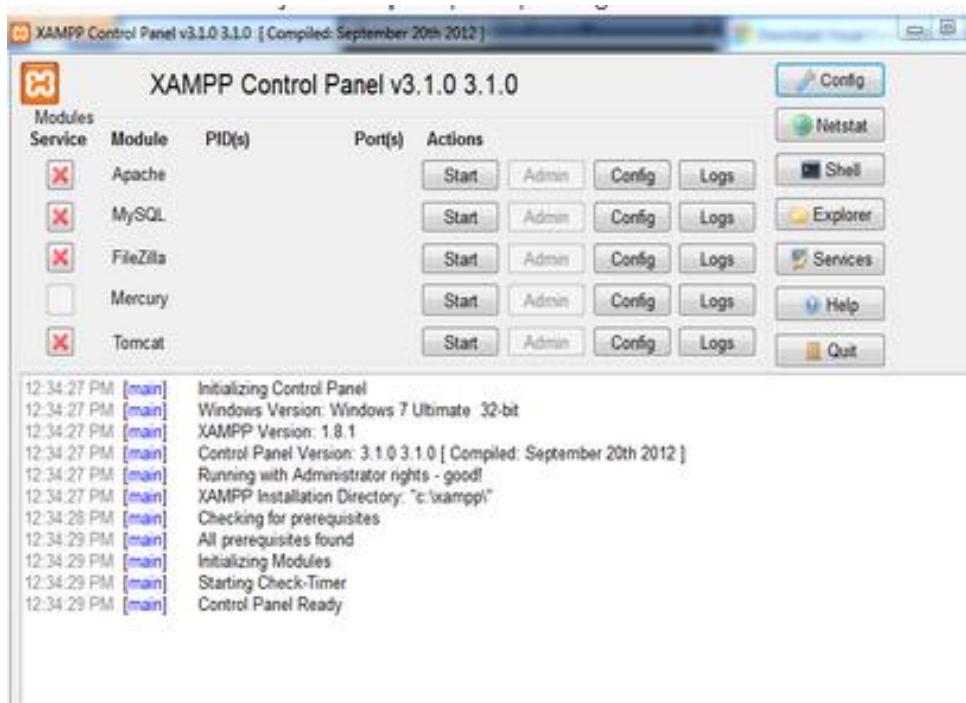
10. Berikutnya, akan muncul jendela dialog seperti gambar di bawah ini yang menanyakan Anda apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik YES.



### 3.1.1.3 Cara Menjalankan XAMPP

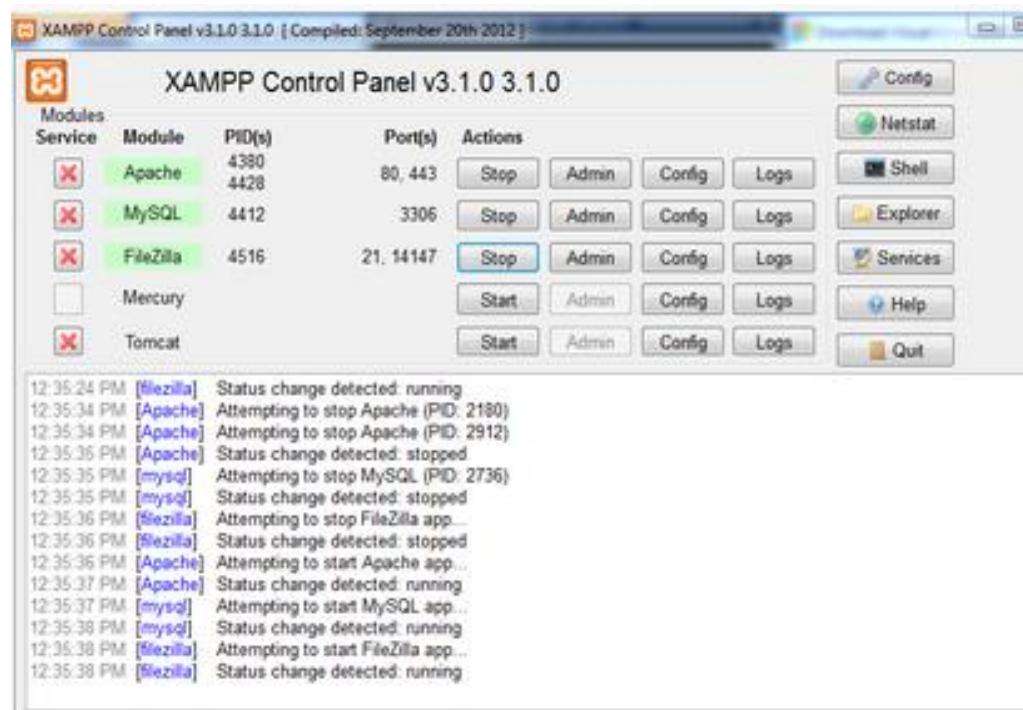
Berikut adalah langkah-langkah untuk menjalankan aplikasi XAMPP :

1. Bukalah aplikasi XAMPP, bisa melalui **Start Menu** atau **Desktop**, dan klik icon XAMPP. Atau, jika Anda membukanya begitu proses instalasi selesai maka klik **Yes** seperti yang terlihat pada gambar di atas.
2. Setelah terbuka, silahkan klik tombol **Start** pada kolom **Action** sehingga tombol tersebut berubah menjadi **Stop**. Dengan mengklik tombol tersebut, artinya itulah aplikasi yang dijalankan. Biasanya jika saya menggunakan XAMPP, yang saya start hanyalah aplikasi Apache dan MySQL, karena saya tidak memerlukan aplikasi seperti Filezilla, dan lain-lain.



Gambar 3. 20 Tampilan Aplikasi XAMPP

Ketika aplikasi sudah dibuka dan muncul seperti tampilan di atas, makna penulis harus menekan button “Start” pada Apache, dan button “Start” pada MySQL.



Gambar 3. 21 Tampilan Setelah Dijalankan

Jika sudah melakukan langkah sebelumnya, makanya akan seperti gambar diatas, tulisan Apache dan MySQLnya akan menjadi hijau setelah dilakukan proses start.

3. Sekarang bukalah browser kesukaan Anda, dan coba ketikkan <http://localhost/xampp> di address bar. Jika muncul tampilan seperti gambar di bawah ini, instalasi telah berhasil.



Gambar 3. 22 Tampilan Pada Browser

Setelah mengikuti kedua langkah sebelumnya maka buka browser yang sering digunakan dan lalu tuliskan perintah <http://localhost/xampp> kemudian “Enter” jika muncul seperti gambar diatas berarti instalasi berhasil dan xampp siap digunakan.

## **BAB IV**

---

### **4.1 PENGENALAN DAN PENJELASAN METODE RAD (RAPID APPLICATION DEVELOPMENT)**

#### **4.1.1 Pengantar Metode RAD**

##### **4.1.1.1 Definisi Metode RAD**

Metode pengembangan sistem adalah suatu aktivitas, metode, praktik terbaik dan peralatan terotomatisasi yang digunakan para *stakeholder* untuk mengembangkan dan secara berkesinambungan memperbaiki sistem informasi dan perangkat lunak (Whitten, Bentley, Ditman, 2004). Pengembangan sistem informasi merupakan penyusunan suatu sistem untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

*Rapid Application Development* (RAD) yaitu suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak (Kendall dan Kendall, 2002).

Agar Anda lebih mengenai bahasa pemrograman PHP dan CSS, kita akan membahas terlebih dahulu apa itu PHP dan apa fungsi PHP di dalam proses pengembangan website.

*Rapid Application Development* (RAD) adalah model proses perkembangan *software* sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis pada komponen (Hariyanto, 2004). Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60 sampai 90 hari).

## **Tahapan-tahapan dalam RAD**

RAD digunakan pada aplikasi sistem konstruksi, maka menekankan fase-fase. Ada tiga fase dalam RAD yaitu (Kendall dan Kendall, 2008):

1. ***Requirement Planning***, dalam tahap ini diketahui apa saja yang menjadi kebutuhan sistem yaitu dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi untuk menentukan tujuan, batasan-batasan sistem, kendala dan juga alternatif pemecahan masalah. Analisis digunakan untuk mengetahui perilaku sistem dan juga untuk mengetahui aktivitas apa saja yang ada dalam sistem tersebut.
2. ***Design Workshop***, yaitu mengidentifikasi solusi alternatif dan memilih solusi yang terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk data-data yang telah didapatkan dan dimodelkan dalam arsitektur sistem informasi. Tools yang digunakan dalam pemodelan sistem biasanya menggunakan Unified Modeling Language (UML).
3. ***Implementation***, setelah *Design Workshop* dilakukan, selanjutnya sistem diimplementasikan (*coding*) ke dalam bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program. Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan.

## **Keunggulan dan Kelemahan Menggunakan RAD**

Berikut adalah keunggulan dan kelemahan menggunakan RAD (Whitten, Bentley, Ditman, 2004):

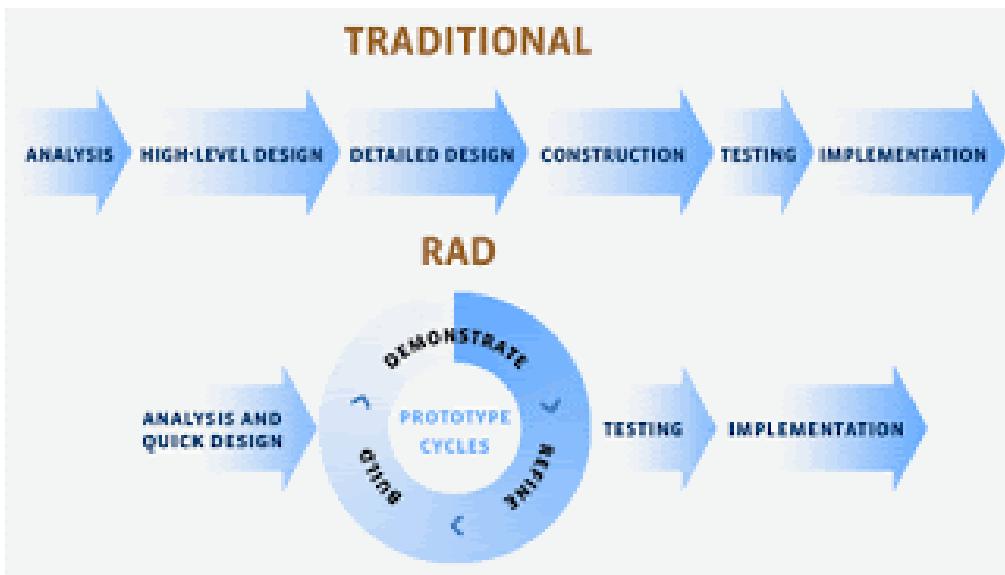
### **Keunggulan**

1. Berguna untuk proyek-proyek tempat persyaratan-persyaratan pengguna tidak pasti dan tidak tepat.

2. Mendorong pengguna aktif dan partisipasi manajemen (berkebalikan dengan reaksi pasif pada model-model sistem yang tidak bekerja). Hal ini meningkatkan antusiasme pengguna akhir pada proyek.
3. Proyek-proyek memiliki visibilitas dan dukungan lebih tinggi karena keterlibatan pengguna yang ekstensif selama proses.
4. Para pengguna dan manajemen melihat solusi-solusi yang berbasis perangkat lunak dan bekerja lebih cepat daripada pengembangan yang *model-driven*.
5. *Error* dan penghilangan cenderung untuk dideteksi lebih awal dan *prototype* daripada dalam model sistem.

## **Kelemahan**

1. Beberapa orang berpendapat bahwa RAD mendorong mentalitas “mengkode, mengimplementasi, dan memperbaiki” yang meningkatkan biaya seumur hidup yang diperlukan untuk mengoperasikan, mendukung dan merawat sistem.
2. *Prototype-prototype* RAD dapat dengan mudah memecahkan yang salah karena analisis masalah disingkat atau diabaikan.
3. *Prototype* berbasis RAD mungkin membuat para analis minder untuk mempertimbangkan alternatif-alternatif teknis lain yang lebih bernilai.
4. Kadang-kadang lebih baik membuang sebuah *prototype*, tapi para *stakeholder* enggan melakukannya karena menganggapnya sebagai hilangnya waktu dan usaha dalam produk saat ini.
5. Penekanan pada kecepatan dapat berdampak terhadap kualitas yang disebabkan jalan-jalan pintas yang disarankan dengan buruk melalui metodologi tersebut.



*Gambar 4. 1 Langkah Penerapan Metode RAD*

#### **4.2.1.2 Sejarah Metode RAD**

*Rapid Application Development (RAD)* adalah istilah awalnya digunakan untuk menggambarkan proses pengembangan perangkat lunak pertama kali dikembangkan dan berhasil digunakan selama pertengahan 1970-an oleh Sistem Pusat Pengembangan New York Telephone Co di bawah arahan Dan Gielan. Setelah serangkaian implementasi sangat berhasil dari proses ini, Gielan kuliah secara ekstensif di berbagai forum pada metodologi , praktek, dan manfaat dari proses ini.

RAD melibatkan pengembangan dan pembangunan prototipe iteratif . Pada tahun 1990 , dalam buku RAD, Rapid Application Development, James Martin didokumentasikan penafsirannya tentang metodologi[3] 17. Baru-baru ini, istilah dan singkatan yang telah datang untuk digunakan dalam lebih luas, pengertian umum yang mencakup berbagai metode yang bertujuan untuk

mempercepat pengembangan aplikasi, seperti penggunaan kerangka perangkat lunak dari berbagai jenis, seperti kerangka kerja aplikasi web.

Pengembangan aplikasi yang cepat merupakan respon terhadap proses yang dikembangkan pada 1970-an dan 1980-an, seperti Structured Sistem Metode Analisis dan Desain dan model Waterfall lainnya. Satu masalah dengan metodologi sebelumnya adalah bahwa aplikasi begitu lama untuk membangun bahwa persyaratan telah berubah sebelum sistem itu selesai, sehingga sistem tidak memadai atau bahkan tidak dapat digunakan. Masalah lain adalah asumsi bahwa persyaratan metodis tahap analisis saja akan mengidentifikasi semua persyaratan penting. Membuktikan fakta bahwa ini adalah jarang terjadi, bahkan untuk proyek-proyek dengan profesional yang sangat berpengalaman di semua tingkatan.

Dimulai dengan ide-ide dari Brian Gallagher, Alex Balchin, Barry Boehm dan Scott Shultz, James Martin mengembangkan pendekatan pengembangan aplikasi yang cepat selama tahun 1980 di IBM dan akhirnya diresmikan itu dengan menerbitkan sebuah buku pada tahun 1991, Rapid Application Development.

#### **4.2.1.3 Contoh Penerapan Metode RAD**

Model RAD mengadopsi model waterfall dan pembangunan dalam waktu singkat yang dicapai dengan menerapkan :

1. Component based construction (pemrograman berbasis komponen bukan prosedural).
2. Penekanan pada penggunaan ulang (reuse) komponen perangkat lunak yang telah ada.
3. Pembangkitan kode program otomatis/semi otomatis.

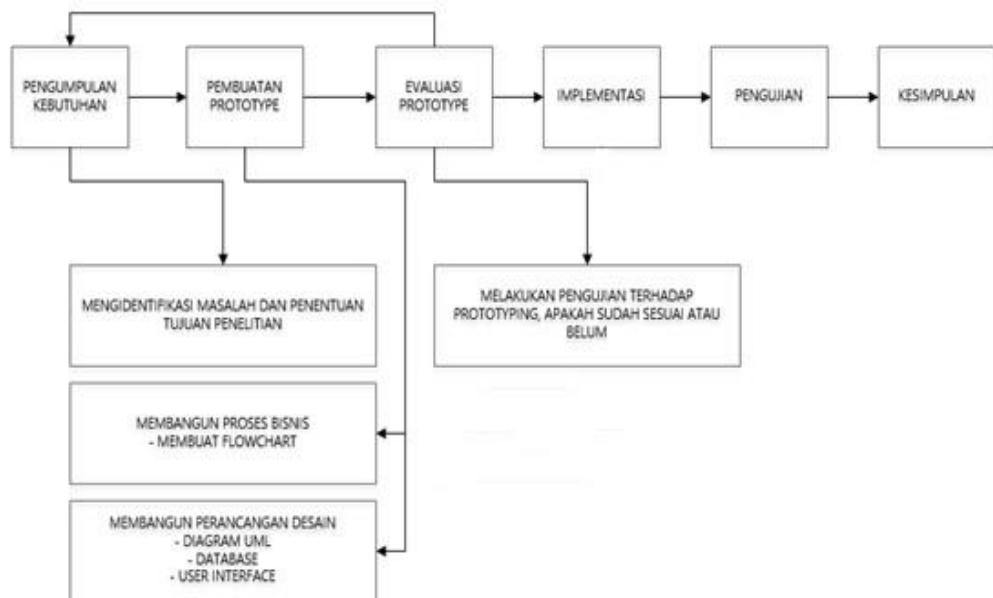
4. Multiple team (banyak tim), tiap tim menyelesaikan satu tugas yang selevel tapi tidak sama. Banyaknya tim tergantung dari area dan kompleksitasnya sistem yang dibangun.

Jika keutuhan yang diinginkan pada tahap analisis kebutuhan telah lengkap dan jelas, maka waktu yang dibutuhkan untuk menyelesaikan secara lengkap perangkat lunak yang dibuat adalah berkisar 60 sampai 90 hari. Model RAD hampir sama dengan model waterfall, bedanya siklus pengembangan yang ditempuh model ini sangat pendek dengan penerapan teknik yang cepat.

Sistem dibagi-bagi menjadi beberapa modul dan dikerjakan beberapa tim dalam waktu yang hampir bersamaan dalam waktu yang sudah ditentukan. Model ini melibatkan banyak tim, dan setiap tim mengerjakan tugas yang selevel, namun berbeda. Sesuai dengan pembagian modul sistem.

## **4.2 Penjelasan Penerapan Metode Dalam Pembuatan Aplikasi E-Office**

#### 4.2.1 DIAGRAM ALUR METODOLOGI PENELITIAN



Gambar 4. 2 Diagram Alur Metodologi Penelitian

#### TAHAPAN – TAHAPAN DIAGRAM ALUR METODOLOGI PENELITIAN

Berikut adalah tahapan – tahapan diagram alur metodologi penelitian berdasarkan gambar sebelumnya :

#### 4.2.2 Pengumpulan Kebutuhan

Pengumpulan kebutuhan dilakukan untuk menentukan permasalahan yang terjadi yang akan diteliti sehingga dapat ditentukan tujuan dari penelitian yang dilakukan, pengumpulan kebutuhan dilakukan dengan cara wawancara dengan pegawai yang akan menggunakan aplikasi ini, dengan wanwancara tersebut peneliti dapat mengetahui apa saja yang menjadi kebutuhan setiap pegawai pada Dinas Perhubungan Provinsi Jawa Barat.

### **4.2.3 Membangun Prototyping**

#### **4.2.3.1 Membangun Proses Bisnis**

Pada tahap ini dibuat proses bisnis atau analisis sistem yang bertujuan mengidentifikasi masalah dan hubungan antarproses yang terlibat dalam sistem, yang meliputi proses bisnis yang sedang berjalan dan proses bisnis yang akan dibangun.

#### **4.2.3.2 Membangun Perancangan Desain**

Pada tahap ini, dibuat rincian sistem agar dapat dimengerti oleh *user* yang terdiri dari tiga tahap, yaitu:

1. Tahap merancang gambaran sistem atau proses yang akan dibangun yang dimodelkan menggunakan diagram UML.
2. Tahap merancang *database* yang akan digunakan.
3. Tahap merancang tampilan *user interface* dan struktur menu sistem yang akan digunakan.

#### **4.2.4 Evaluasi Prototyping**

Setelah melakukan pembangun *prototyping*, dilakukan evaluasi dalam bentuk persentasi pada semua pegawai yang akan menggunakan aplikasi ini untuk mengetahui apakah *prototyping* yang akan digunakan sudah sesuai dengan kebutuhan atau belum. Jika sudah sesuai, maka akan lanjut ke langkah selanjutnya. Dan apabila belum sesuai, maka akan direvisi dan kembali lagi untuk mengumpulkan kebutuhan.

#### **4.2.5 Implementasi**

Implementasi merupakan tahap pembuatan sistem berdasarkan hasil perancangan dan pengumpulan kebutuhan yang telah dibuat sebelumnya sehingga sistem dapat difungsikan dalam keadaan yang sebenarnya dan dapat diketahui apakah sistem yang dibuat sudah berhasil mencapai tujuan yang sebenarnya atau belum.

#### **4.2.6 Pengujian**

Setelah implementasi sistem selesai, langkah selanjutnya adalah pengujian terhadap sistem yang telah dibuat menggunakan pengujian *Black-box*. Pengujian bertujuan mengetahui apakah sistem yang telah dibuat sudah sesuai dengan kebutuhan *user* dan mengetahui apakah fungsi *input* dan *output* sistem sudah sesuai dengan yang diharapkan atau tidak.

#### **4.2.7 Kesimpulan dan Saran**

Dari hasil pengujian sistem yang telah dibuat, maka penulis dapat membuat kesimpulan dari hasil penelitian yang dilakukan sehingga penulis bisa memberikan saran terhadap sistem yang telah dibuat.

## **BAB V**

### **5.1 ANALISIS DAN PERANCANGAN SISTEM**

---

Analisis adalah aktivitas yang terdiri dari serangkaian kegiatan seperti, mengurai, membedakan, memilah sesuatu untuk dikelompokkan kembali menurut kriteria tertentu dan kemudian dicari kaitannya lalu ditafsirkan maknanya.

Analisis dapat juga diartikan sebagai usaha dalam mengamati sesuatu secara mendetail dengan cara menguraikan komponen-komponen pembentuknya atau menyusun komponen tersebut untuk dikaji lebih lanjut.

Ada juga yang menganggap arti analisis sebagai kemampuan dalam memecahkan atau menguraikan suatu informasi atau materi menjadi komponen-komponen yang lebih kecil sehingga lebih mudah dimengerti dan mudah dijelaskan.

Kata analisis banyak digunakan dalam berbagai bidang ilmu pengetahuan, baik itu pengetahuan sosial, manajemen, ekonomi bisnis, akuntansi, ilmu bahasa, pengetahuan alam, dan bidang ilmu lainnya.

Kata analisis atau analisa berasal dari bahasa Yunani Kuno, yaitu “analusis” yang artinya melepaskan. Beberapa ahli pernah menjelaskan mengenai arti analisis, diantaranya adalah:

#### **1. Komaruddin**

Menurut Komaruddin, pengertian analisis adalah aktivitas berfikir untuk menguraikan suatu keseluruhan menjadi komponen-komponen kecil sehingga dapat mengenal tanda-tanda komponen, hubungan masing-masing

komponen, dan fungsi setiap komponen dalam satu keseluruhan yang terpadu.

## **2. Wiradi**

Menurut Wiradi, arti analisis adalah aktivitas yang memuat kegiatan memilah mengurai, membedakan sesuatu yang kemudian digolongkan dan dikelompokkan menurut kriteria tertentu lalu dicari makna dan kaitannya masing-masing.

## **3. Dwi Prastowo Darminto**

Menurut Dwi Prastowo Darminto, pengertian analisis adalah penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian itu sendiri, serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan.

## **4. Syahrul**

Pengertian analisis dalam akuntansi menurut Syahrul adalah kegiatan melakukan evaluasi terhadap kondisi dari pos-pos atau ayat-ayat yang berkaitan dengan akuntansi dan alasan-alasan yang memungkinkan tentang perbedaan yang muncul.

## **5. Rifka Julianty**

Menurut Rifka Julianty, pengertian analisis adalah aktivitas penguraian pada pokok atas berbagai bagiannya dan penelaahan bagian itu

sendiri, serta hubungan antara bagian untuk mendapatkan pengertian yang tepat dan pemahaman arti keseluruhan.

## **6. Mohammad Afdi Nizar**

Menurut Mohammad Afdi Nizar, pengertian analisis dalam akuntansi adalah evaluasi mengenai kondisi dari ayat-aya yang berhubungan dengan akuntansi dan alasan yang memungkinkan sebuah perbedaan akan muncul.

## **7. Robert J. Schreiter**

Menurut Robert J. Schreiter pengertian analisis adalah “membaca” teks yang melokalisasikan berbagai tanda dan menempatkan tanda-tanda tersebut dalam interaksi yang dinamis, dan pesan-pesan yang disampaikan.

## **8. Minto Rahayu**

Menurut Minto Rahayu, arti analisis adalah sebuah cara dalam membagi suatu subyek ke dalam komponen-komponen, meliputi melepaskan, menanggalkan, menguraikan sesuatu yang terikat padu.

## **9. Husein Umar**

Menurut Husein Umar, pengertian analisis adalah suatu proses kerja dari rangkaian tahapan pekerjaan sebelum riset, didokumentasikan dengan tahapan pembuatan laporan.

## **10. Efrey Liker**

Menurut Efrey Liker, arti analisis adalah aktivitas dalam mengumpulkan bukti, untuk menemukan sumber suatu masalah, yaitu akarnya.

## **11. Kamus Besar Bahasa Indonesia (KBBI)**

Pengertian analisis menurut KBBI adalah:

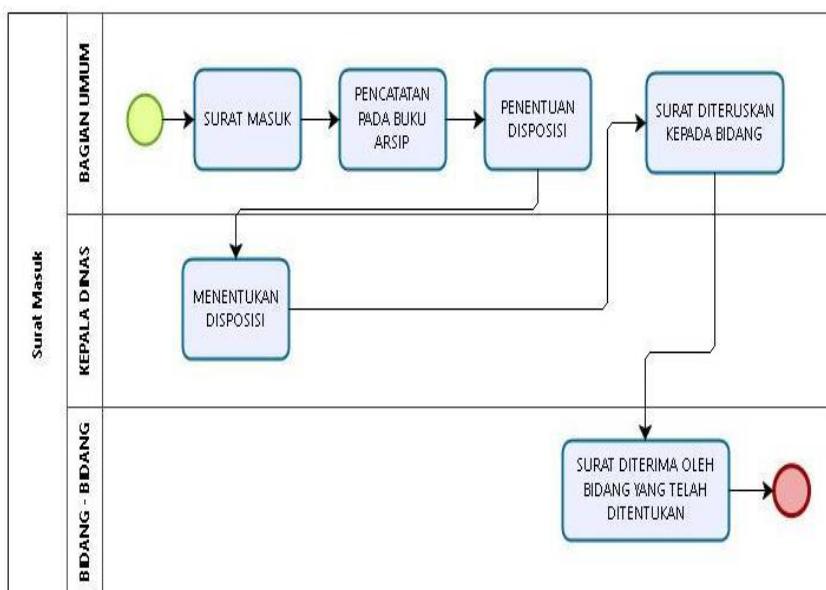
- Penyelidikan terhadap suatu peristiwa (karangan, perbuatan, dan lainnya) untuk mengetahui keadaan yang sebenarnya (sebab musabab, duduk perkara, dan sebagainya).
- Aktivitas penguraian suatu pokok atas berbagai baginya dan penelaahan bagian itu sendiri serta hubungan antarbagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan
- Pemecahan persoalan yang dimulai dengan dugaan akan kebenarannya
- Penjabaran sesudah dikaji sebaik-baiknya

### **5.1.1 Analisis yang berjalan**

Tahapan yang di perlukan dalam pembuatan suatu program yaitu menganalisa sistem yang telah ada, dimana analisa sistem merupakan proses mempelajari suatu sistem dengan cara menguraikan sistem tersebut kedalam elemen yang membentuknya. Selanjutnya mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang terjadi serta kebutuhan yang diperlukan, sehingga dapat di usulkan pembuatannya.

### 5.1.1.1 Analisis Prosedur yang berjalan

#### 1. Analisis Prosedur Surat Masuk Yang Sedang Berjalan



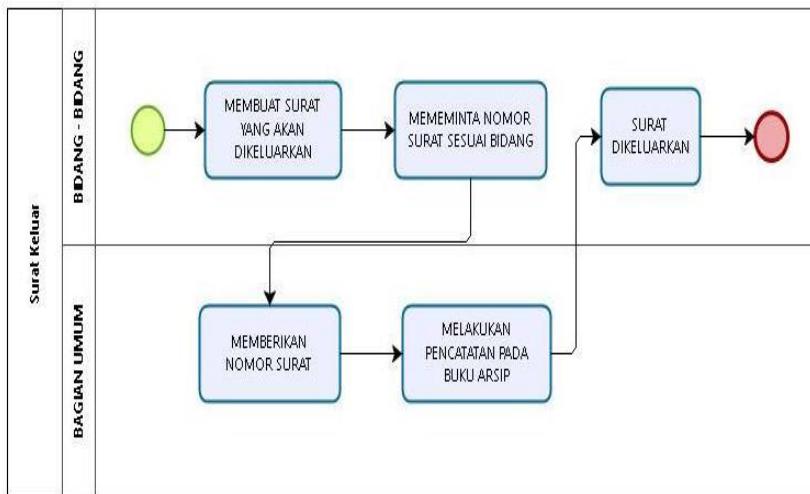
Powered by  
**bizagi**  
Modeler

Gambar 5. 1 BPMN Prosedur Surat Masuk yang Sedang Berjalan

Pada gambar 5.1 BPMN di atas adalah penjelasan alur yang berjalan pada proses surat masuk dinas perhubungan. Mulai dari menerima surat, setelah itu

pencatatan pada buku arsip, penentuan disposisi oleh Kepala Dinas Setelah itu surat diteruskan kepada bidang yang sesuai dengan disposisi yang sudah ditentukan. Lalu bidang yang di tuju akan menerima surat.

## 2. Analisis Prosedur Surat Keluar Yang Sedang Berjalan



Powered by  
**bizagi**  
Modeler

Gambar 5. 2 BPMN Prosedur Surat Keluar yang Sedang Berjalan

Pada gambar 5.2 BPMN diatas adalah penjelasan alur yang berjalan pada proses surat keluar dinas perhubungan. Mulai dari pembuatan surat, meminta nomor surat sesuai bidang, melakukan pencatatan pada buku arsip. Lalu surat dikeluarkan.

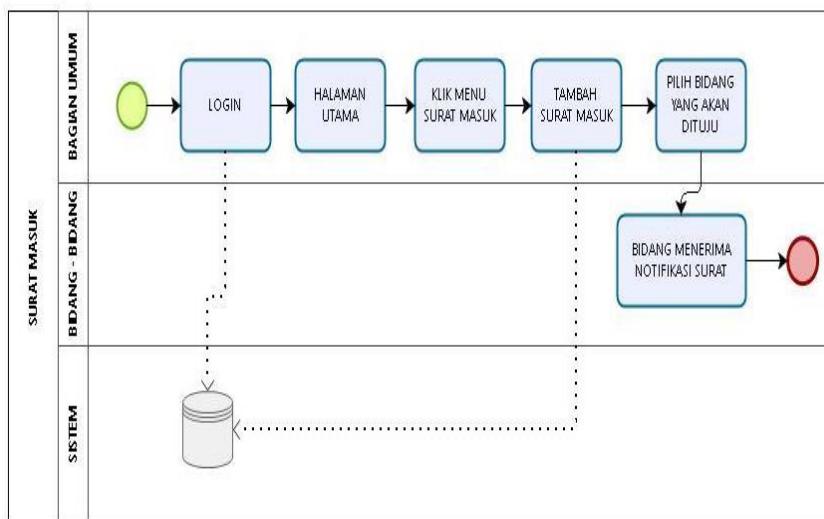
### 5.1.2 Analisis yang akan dibangun

Dalam membangun sistem perlu dibuat sebuah perancangan. Perancangan sistem ini ditujukan agar dalam pembentukan sistem dapat

dihasilkan suatu aplikasi yang berfungsi secara optimal dan memberi kemudahan bagi penggunanya.

### 5.1.2.1 Analisis Prosedur yang akan dibangun

#### 1. Analisis BPMN Surat Masuk Yang Akan Dibangun



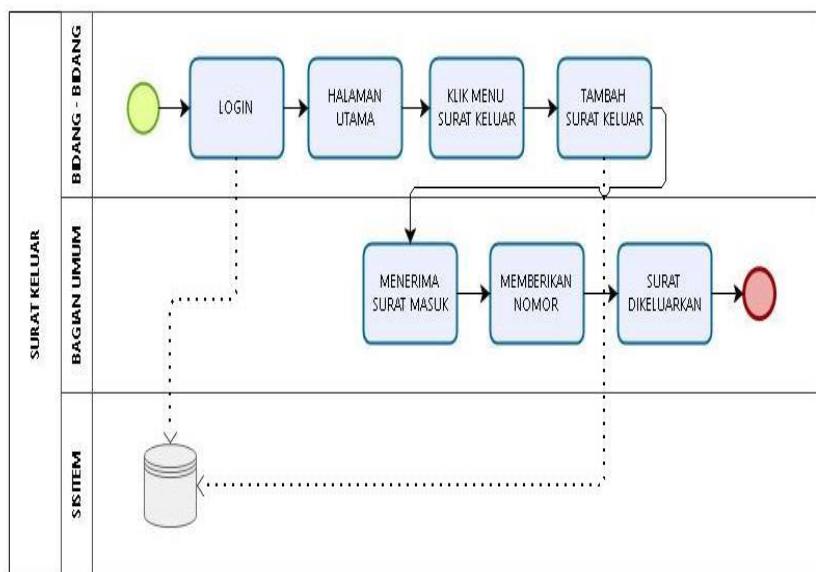
Powered by  
**bizagi**  
Modeler

Gambar 5. 3 BPMN Surat Masuk Yang Akan Dibangun

Pada gambar 5.3 BPMN diatas adalah menjelaskan alur proses diagram sistem yang akan dibangun pada surat masuk yaitu bagian umum melakukan login terlebih dahulu, jika valid maka akan masuk ke halaman utama jika tidak valid maka akan kembali lagi ke halaman login. Lalu bagian umum menginput surat masuk pada menu surat masuk dan setelah itu disimpan ke dalam database. Lalu bidang - bidang melakukan login

untuk masuk ke halaman bidang. Dan bidang - bidang menerima surat masuk.

## 2. Analisis BPMN Surat Keluar Yang Akan Dibangun



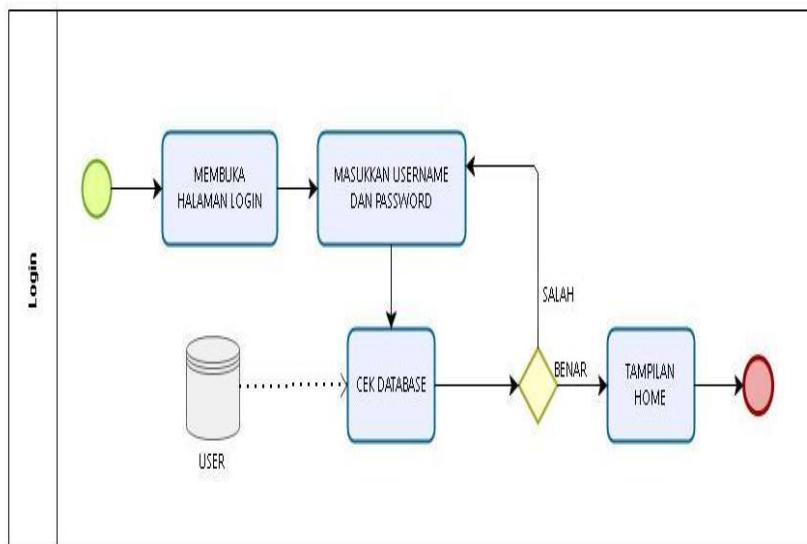
Powered by  
**bizagi**  
Modeler

Gambar 5. 4 BPMN Surat Keluar Yang Akan Dibangun

Pada gambar 5.4 BPMN diatas adalah menjelaskan alur proses diagram sistem yang akan dibangun pada surat keluar yaitu bidang melakukan login terlebih dahulu, jika valid maka akan masuk ke halaman utama jika tidak valid maka akan kembali lagi ke halaman login. Lalu bidang menginput surat keluar pada menu surat keluar dan setelah itu

disimpan ke dalam database. Lalu bagian umum melakukan login untuk masuk ke halaman bagian umum. Dan bagian umum menerima surat keluar dan memberikan nomor, kemudian surat dikeluarkan.

### 3. Analisis BPMN Yang Akan Dibangun Pada *Login*



Powered by  
**bizagi**  
Modeler

Gambar 5. 5 BPMN Yang Akan Dibangun Pada *Login*

Pada prosedur *login*, *user* akan masuk dengan menggunakan *username* dan *password* lalu akan dicek oleh sistem apakah masukan yang diberi oleh *user* sesuai atau tidak. Bila sesuai maka sistem akan otomatis beralih ke halaman home sesuai dengan hak aksesnya. Bila tidak sesuai maka *user* diharuskan mengisi kembali *username* dan *passwordnya*.

## **5.2 Perancangan**

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengonfigurasi komponen – komponen perangkat lunak dan perangkat keras dari suatu sistem.

Konsep merancang multimedia merupakan aplikasi multimedia yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi multimedia dibutuhkan kreatifitas. Kreatifitas adalah kemampuan untuk menyajikan gagasan atau ide baru. Sedangkan inovasi merupakan aplikasi dari gagasan atau ide baru tersebut. Untuk menciptakan ide yang orisinal tidaklah mudah, maka dapat digunakan beberapa teknik untuk menciptakan ide, yaitu penyesuaian (adaptasi), Multimedia yang telah ada dianggap belum sesuai dengan lingkungan yang dituju. Merancang konsep analisis sistem bekerjasama dengan pemakai, mungkin juga bekerjasama dengan profesional komunikasi seperti produser, sutradara, penulis naskah, editor elektronik terlibat dalam merancang konsep yang menentukan keseluruhan pesan dan membuat aliran (urutan) pada 12 aplikasi multimedia yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi multimedia dibutuhkan kreativitas. Multimedia yang telah ada dianggap belum sesuai dengan lingkungan yang dituju walaupun isinya telah sesuai dengan lingkungan. Misalnya multimedia ditujukan ke kalangan kawula muda

namun multimedia yang ada hanya sesuai untuk kalangan orang tua (dari sisi desain tampian, bahasa dll) pembesaran (maksimasi), Multimedia yang ada dianggap sangat sederhana, sehingga perlu untuk dikembangkan lebih kompleks pengecilan (minimasi), Multimedia yang ada dianggap terlalu rumit dan sulit untuk difahami, dimengerti, dioperasikan dll. Sehingga perlu untuk dilakukan. Penyederhanaan pembalikan (inversi), Multimedia yang telah ada dianggap memiliki isi yang keliru, sehingga diperlukan perubahan isi multimedia secara menyeluruh terhadap pengembangan multimedia tersebut perubahan (modifikasi), pengaturan kembali dan perpaduan (kombinasi). Multimedia dianggap belum sesuai dengan kebutuhan dan dirubah dengan menggunakan teknik perubahan lebih dari satu cara pengembangan.

### **5.2.1 Perancangan UML**

UML merupakan singkatan dari “*Unified Modelling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software.

### **B. Tujuan atau fungsi dari penggunaan UML**

Inilah beberapa tujuan atau fungsi dari penggunaan UML, yang diantaranya:

- Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.

- Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
- Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
- Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
- Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
- Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

Perancangan merupakan penjelasan perancangan sistem/aplikasi/alat yang akan dibuat terdiri dari perancangan alir program (*flow chart*), algoritma, data, maupun perancangan input/output sistem/aplikasi/alat. Perancangan yang akan dibuat yaitu menggunakan object oriented UML terdiri dari *use case diagram*, *class diagram*, *sequence diagram*, *collaboration diagram*, *activity diagram*, *statechart diagram*, *component diagram*, *deployment diagram*, *objek diagram*, *structure program*, dan perancangan *user interface*.

#### **5.2.1.1   *Use Case Diagram***

Use-case diagram merupakan model diagram UML yang digunakan untuk menggambarkan requirement fungsional yang

diharapkan dari sebuah sistem. Usecase diagram adalah diagram usecase yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Use case class digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (or bagian sistem: subsistem atau class) ke pemakai. Diagram use case tidak menjelaskan secara detail tentang penggunaan usecase, namun hanya memberi gambaran singkat hubungan antara usecase, aktor, dan sistem.

Melalui diagram usecase dapat diketahui fungsi-fungsi apa saja yang ada pada sistem (Rosa-Salahudin, 2011: 130). Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use Case Diagram kerap digunakan untuk mendokumentasikan dan menjelaskan proses-proses yang berlangsung di dalam sebuah sistem. Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Use case Diagram, adalah gambaran efek fungsionalitas yang diharapkan oleh sistem.

### Tujuan Use Case

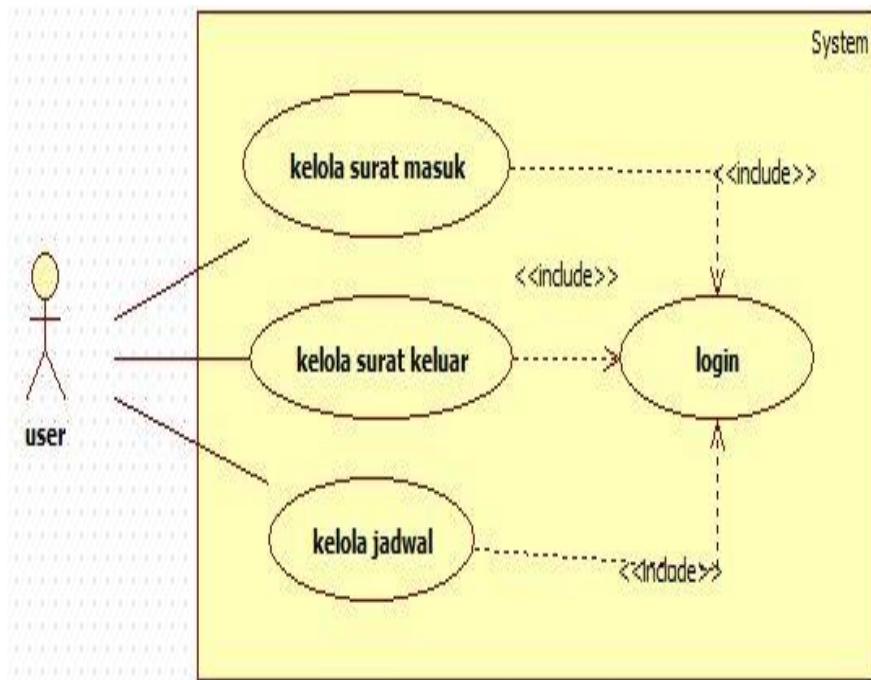
- Memetakkan kebutuhan sistem

- Merepresentasikan interaksi pengguna terhadap sistem
- Untuk mengetahui kebutuhan diluar sistem

### **Deskripsi Use Case**

- Diagram use case merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat.
- Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.
- Diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.
- Sebuah use case merepresentasikan sebuah interaksi antara aktor (user atau sistem lainnya) dengan sistem.
- Use case menjelaskan secara sederhana fungsi sistem dari sudut pandang user.

*Use case diagram* merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antara aktor dengan aktivitas yang terjadi pada sistem dan alat. berikut ini menggambarkan tentang proses yang terjadi secara umum:



*Gambar 5. 6 Use Case Diagram.*

#### 5.2.1.1.1 Definisi Aktor

*Tabel 5. 1 Definisi Aktor*

<b>o</b>	<b>ktor</b>	<b>eterangan</b>
	user	elakukan kelola surat masuk, surat keluar, i, dan jadwal / schedule.

### **5.2.1.1.2 Definisi Use Case**

*Table 5. 1 Definisi Use Case*

<b>o</b>	<b>use Case</b>	<b>eskripsi</b>
	<i>login</i>	Melakukan proses <i>login</i>
	elola Surat Masuk	terupakan serangkaian aktivitas dilakukan untuk mengelola surat
	elola Surat Keluar	terupakan serangkaian aktivitas dilakukan untuk mengelola surat
	elola Disposisi	terupakan serangkaian aktivitas dilakukan untuk mengelola i surat.
	elola Jadwal / Schedule	terupakan serangkaian aktivitas dilakukan untuk mengelola jadwal edule pegawai.

### **5.2.1.1.3 Use case Diagram Skenario**

*Use Case Skenario* mendeskripsikan urutan langkah-langkah dalam proses sistem, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut ini penjelasan dari masing-masing skenario tersebut :

### 5.2.1.1.3.1 Skenario Use Case Login User

Table 5. 2 Skenario Use Case Login User

Identifikasi	
<b>Nomor</b>	C1
<b>Nama</b>	login
<b>ujuan</b>	emberikan hak akses <i>user</i> ke sistem dengan melakukan verifikasi terhadap <i>username</i> dan <i>password</i> yang dimasukan oleh <i>user</i> .
eskripsi	
<b>Aktor</b>	user
Skenario Utama	
<b>Kondisi Awal</b>	Form Login sudah tersedia
<b>Kksi Aktor</b>	eaksi Sistem
Memasukkan <i>username</i> dan <i>password</i>	Form Login akan dipilkan <i>textbox</i> <i>username</i> , <i>password</i> dan untuk <i>password</i> akan dalam bentuk kode ‘*’ di layar untuk jaminan keamanan.

Aktor melakukan konfirmasi uan terhadap <i>username</i> , <i>password</i> lah dimasukkan dengan menekan <i>Login</i> .	Aplikasi melakukan terhadap <i>username</i> , <i>password</i> yang telah dimasukkan ngguna dengan melakukan kan pada basis data.
<b>ondisi Akhir</b>	ka pada akhir interaksi <i>username</i> , <i>password</i> yang akan pengguna <i>valid</i> maka ia akan langsung masuk ke utama.

### 5.2.1.1.3.3 Skenario Use Case Kelola Surat Masuk

Table 5. 3 Skenario Use Case Kelola Data User

Identifikasi	
<b>nomor</b>	C3
<b>nama</b>	elola Surat Masuk
<b>ujuan</b>	elakukan Imputan Surat
eskripsi	
<b>ktor</b>	ser
kenario Utama	
<b>ondisi Awal</b>	<i>form login</i> terlebih dahulu. <i>lid</i> maka masuk ke <i>form</i> utama jika tidak <i>valid</i> embali ke tampilan awal
<b>ksi Aktor</b>	<b>eaksi Sistem</b>
Pengguna memilih <i>link</i> ke halaman masuk.	Aplikasi menampilkan sesuai pilihan pengguna.
Pengelolaan isi <i>record</i> dengan tombol tambah, ubah atau hapus.	Aplikasi menyimpan lola pada basis data tabel suk atau yang terkait..

<b>ondisi Akhir</b>	i informasi <i>record</i> pada ata tabel data masuk yang an pengelolaan mengalami an.

#### 5.2.1.3.4 Skenario Use Case Kelola Surat Keluar

Table 5. 4 Skenario Use Case Kelola Surat Keluar

Identifikasi	
<b>omor</b>	C4
<b>ama</b>	elola Surat Keluar
<b>ujuan</b>	elakukan Imputan Surat
eskripsi	
<b>ktor</b>	ser
kenario Utama	
<b>ondisi Awal</b>	<i>orm login</i> terlebih dahulu. <i>lid</i> maka masuk ke <i>orm</i> utama jika tidak <i>valid</i> embali ke tampilan awal
ksi Aktor	eaksi Sistem
Pengguna memilih <i>link</i> ke halaman luar.	Aplikasi menampilkan n sesuai pilihan pengguna.

Pengelolaan isi <i>record</i> dengan tombol tambah, ubah atau hapus.	Aplikasi menyimpan lola pada basis data tabel uar atau yang terkait..
<b>ondisi Akhir</b>	i informasi <i>record</i> pada ta tabel data keluar yang an pengelolaan mengalami an.

#### 5.2.1.1.3.6 Skenario Use Case Kelola Jadwal / Schedule

Table 5. 5 Skenario Use Case Kelola Jadwal / Schedule

identifikasi	
<b>nomor</b>	C6
<b>nama</b>	elola Jadwal
<b>ujuan</b>	elakukan penentuan schedule pegawai
eskripsi	
<b>aktor</b>	ser
skenario Utama	
<b>ondisi Awal</b>	<i>form login</i> terlebih dahulu. <i>valid</i> maka masuk ke <i>form utama</i> jika tidak <i>valid</i>

	Kembali ke tampilan awal
Aksi Aktor	Aksi Sistem
Pengguna memilih <i>link</i> ke halaman schedule.	Aplikasi menampilkan sesuai pilihan pengguna.
Pengelolaan isi <i>record</i> dengan tombol tambah, ubah atau hapus.	Aplikasi menyimpan lola pada basis data tabel atau yang terkait..

### 5.2.2 Class Diagram

*Class Diagram* menggambarkan *class* dan hubungan antar *class* didalam sistem. *Class* digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian paling atas diisikan nama *class*, bagian tengah diisikan *variabel* yang dimiliki *class*, dan bagian bawah diisikan *method-method* dari *class*.

**Class Diagram** adalah salah satu jenis diagram yang paling berguna di UML, hal ini karena dapat dengan jelas memetakan struktur sistem tertentu dengan memodelkan kelas, atribut, operasi serta hubungan antar objek.

Class Diagram menggambarkan serta deskripsi atau penggambaran dari class, atribut, dan objek disamping itu juga hubungan satu sama lain seperti pewarisan, containmet, asosiasi dan lainnya.

Class Diagram mampu memberikan kita pandangan yang lebih luas mengenai suatu sistem dengan cara menunjukkan kelas serta hubungan-hubungannya. Diagram class dapat dikatakan bersifat statis, alasannya karena diagram kelas tidak menggambarkan apa yang terjadi jika mereka berhubungan melainkan menggambar hubungan apa yang terjadi.

### **Manfaat Class Diagram**

Class Diagram menawarkan banyak manfaat untuk siapapun, class diagram UML bermanfaat untuk

1. Menggambarkan suatu model data untuk sistem informasi, tidak peduli apakah model data tersebut sederhana maupun kompleks.
2. Dengan mempelajari class diagram maka akan meningkatkan pemahaman mengenai gambaran umum skema dari suatu aplikasi.
3. Mampu menyatakan secara visual akan kebutuhan spesifik suatu informasi serta dapat menyebarkan informasi tersebut ke seluruh bisnis.

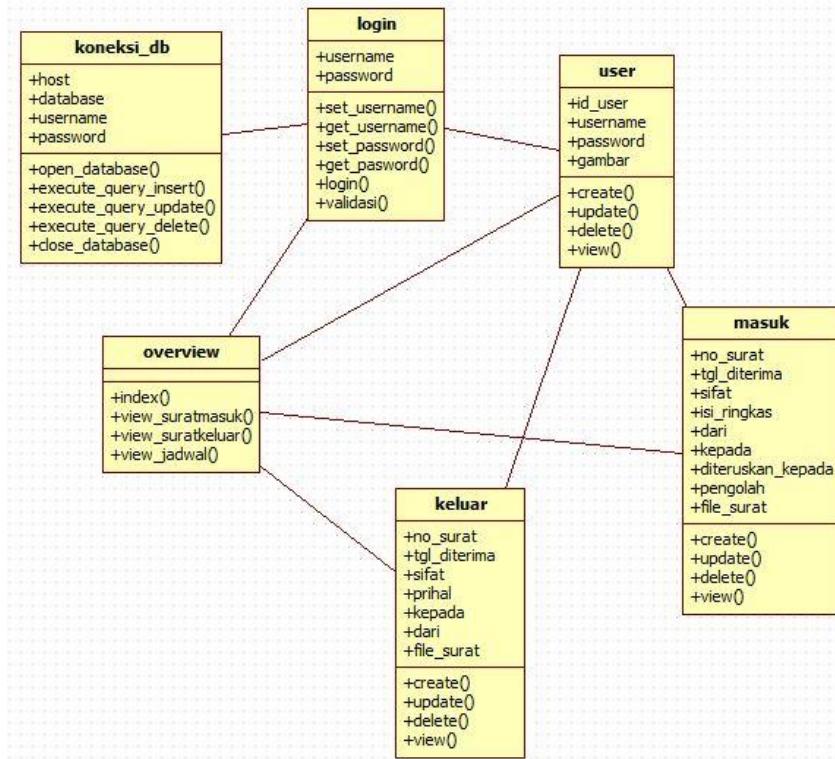
4. Dengan Class Diagram dapat dibuat bagan secara jelas dan terperinci dengan cara memperhatikan kode spesifik apa saja yang dibutuhkan suatu program sehingga mampu mengimplementasikannya ke struktur yang digambarkan.

5. Class Diagram mampu memberikan penggambaran implementasi-independen dari suatu jenis sistem yang digunakan, kemudian dilewatkan diantara berbagai komponen-komponennya.

<b>Komponen</b>	<b>Dasar</b>	<b>Class</b>	<b>Diagram</b>
-----------------	--------------	--------------	----------------

Class diagram terdiri atas tiga bagian atau 3 area pokok, yaitu :  
A) Bagian atas: Berisi **Nama** kelas, bagian ini selalu diperlukan, baik ketika berbicara tentang classifier atau objek.  
Setiap class memiliki nama yang membedakan dengan class yang lain, nama dikenal sebagai simple name (nama sederhana) dan path name (nama path).

Dalam proses pembuatan aplikasi ini *class diagram* yang akan digunakan adalah sebagai berikut :



*Gambar 5. 7 Class Diagram.*

### **5.2.3 Sequence Diagram**

*Sequence diagram* adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram, pada diagram ini, dimensi vertikal merepresentasikan waktu.

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek – obyek yang melakukan suatu tugas atau aksi tertentu. Obyek – obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Pada diagram ini, dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis Vertical, disebut lifeline, dilekatkan pada setiap obyek atau aktor. Kemudian, lifeline tersebut digambarkan menjadi kotak ketika obyek melakukan suatu operasi , kotak tersebut disebut activation box. Obyek dikatakan mempunyai live activation pada saat tersebut.Pesan yang dipertukarkan antar obyek digambarkan sebagai sebuah anak panah antara activation box pengirim dan penerima. Kemudian diatasnya diberikan label pesan.

Tujuan penggunaan sequence diagram :

1. Mengkomunikasikan requirement kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model design.
2. Merupakan diagram yang paling cocok untuk mengembangkan model deskripsi use-case menjadi spesifikasi design.

Waktu Penggunaan sequence diagram :

1. Pada tahap discovery : Sequence diagram dari use case yang dipilih dibuat oleh Business Analyst.
2. Pada tahap construction : System analyst menggunakan use case dan sequence diagram yang dibuat oleh Business Analyst sebagai input untuk pembuatan technical sequence diagram yang akan ditinjau kembali oleh Business Analyst
3. Pada tahap final verification and validation : sequence diagram digunakan untuk merancang test dari use-case scenario.

Elemen – elemen sequence diagram :

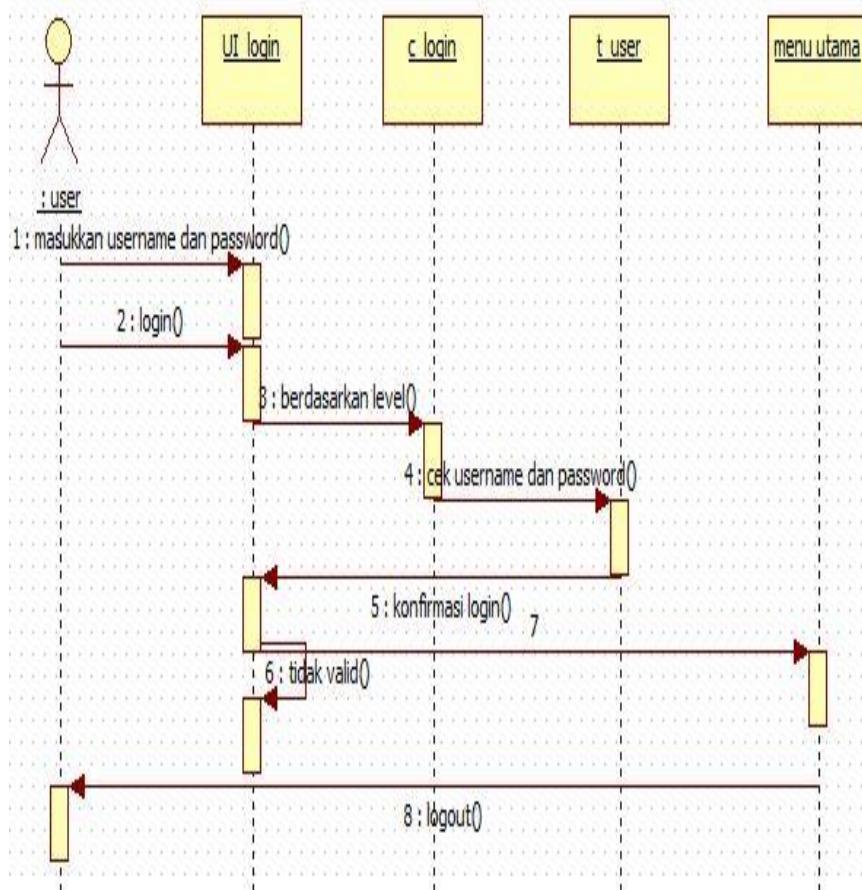
1. Actor
2. Lifeline
3. Boundary
4. Control
5. Entity
6. Fragment
7. End point
8. Diagram gate
9. State

## **B. Listing, Logika dan Output dari Program**

Pada kesempatan kali ini, saya akan menjelaskan sedikit tentang pembuatan sequence diagram yang saya buat. Sequence diagram yang saya buat mengenai “ Prosedur Masuk Area Parkir “. Mengapa saya membuat sequence diagram prosedur masuk area parkir? Karena pada saat – saat ini banyak sekali di tempat – tempat umum yang sudah menggunakan mesin dengan otomatis. Pada mesin ini akan mengeluarkan sebuah karcis yang dimana karcis tersebut biasanya berisikan beberapa hal yang berada di kendaraan. Misalnya : nomor polisi kendaraan, waktu masuk, jenis kendaraan. Disinilah saya akan menjelaskannya. Agar lebih jelas saya akan memberikan contoh dengan nama “ Pengendara “. Pengendara ini merupakan seorang pengendara kendaraan bermotor. Disini pengendara akan memasuki parkiran, mengambil karcis, dan memakirkan kendaraannya di area parkir.

Pertama – tama pengendara sudah jelas harus memasuki kendaraannya tersebut ke tempat parkir. Akan tetapi, sebelum memasuki parkiran petugas parkir akan mencatat nomor polisi, jenis kendaraan dan waktu masuk kendaraan tersebut. Petugas parkir akan membut database atas kendaraan yang masuk ke area parkir. Setelah database dibuat, petugas memerintahkan mesin karcis untuk mencetak dan petugas akan memberikan karcis yang sudah di cetak ke pengendara kendaraan tersebut. Setelah memberikan karcis maka petugas parkir membuka palang dan palang akan terbuka secara otomatis dan pengendara memasuki area parkir dan pengendara memakirkan kendaraannya di area parkir.

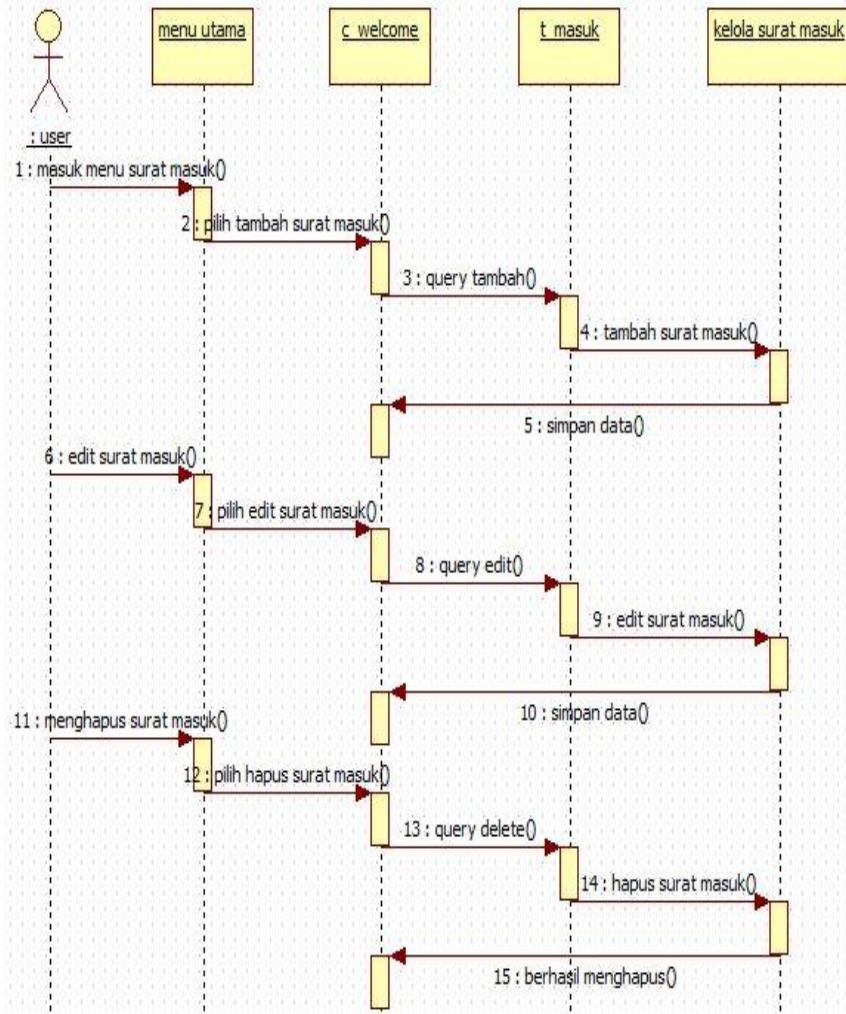
### 5.2.3.1 Sequence Diagram Login User



Gambar 5. 8 Sequence Diagram Login

Pada *sequence diagram* ini menjelaskan proses *login*. Bagian Umum mulai menjalankan aplikasi akan tampil *form login*. Kemudian bagian umum memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di *table user*. Jika sesuai maka akan mendapatkan tampilan halaman utama bagian umum.

### 5.2.3.3 Sequence Diagram Kelola Surat Masuk

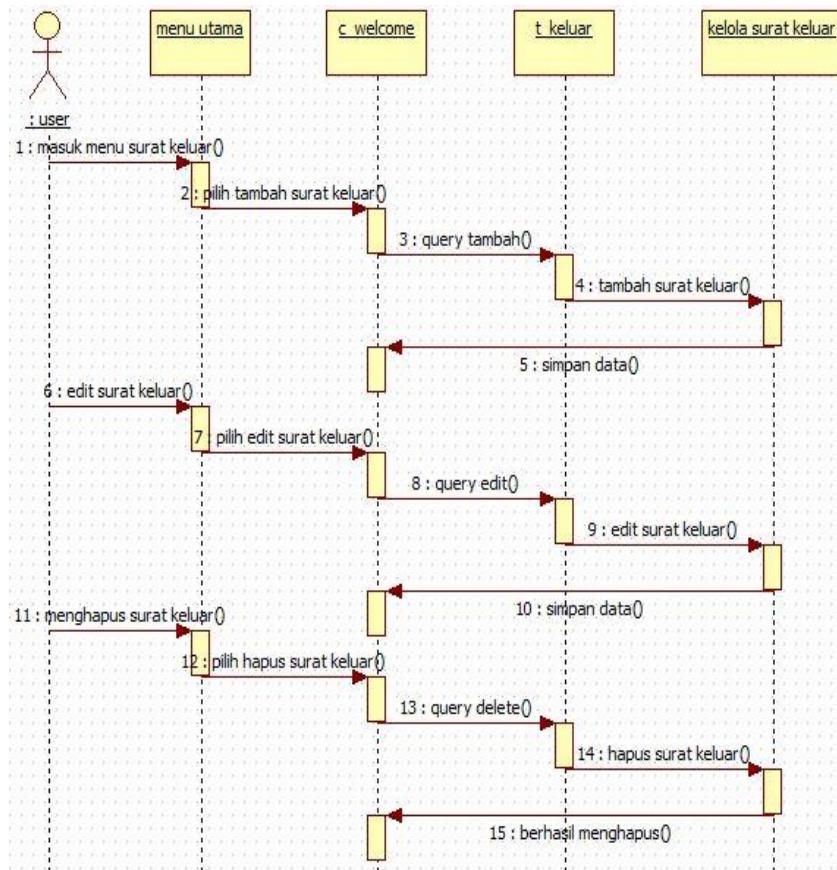


Gambar 5. 9 Sequence Diagram Kelola Surat Masuk

Pada *sequence diagram* ini menjelaskan proses mengelola data *user* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu kelola data *user*, kemudian sistem akan melakukan koneksi

*database*. Setelah *form* tampil, bagian umum dan bidang dapat melakukan operasi tambah data, edit data dan hapus data. Selanjutnya sistem akan menyimpan kembali pada *database*.

#### 5.2.3.4 Sequence Diagram Kelola Surat Keluar



Gambar 5. 10 Sequence Diagram Kelola Surat keluar

Pada *sequence diagram* ini menjelaskan proses mengelola data *user* melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form*

menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu kelola data user, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, bagian umum dan bidang dapat melakukan operasi tambah data, edit data dan hapus data. Selanjutnya sistem akan menyimpan kembali pada *database*.

#### **5.2.4 Collaboration Diagram**

*Collaboration Diagram* adalah suatu diagram yang memperlihatkan atau menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya sequence diagram) dan hubungannya terhadap yang lainnya. Collaboration Diagram lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/message.

*Collaboration diagram* adalah diagram yang mengelompokkan pesan pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram tersebut terdapat method yang dijalankan antara objek yang satu dan objek lainnya. Di diagram kolaborasi ini, objek harus melakukan sinkronisasi pesan dengan serangkaian pesan-pesan lainnya.

*Collaboration Diagram* juga dikenal sebagai **Communication Diagram** atau **Interaction Diagram**, merupakan ilustrasi dari relasi dan interaksi antara objek software pada Unified Modeling Language (UML).

Collaboration Diagram mengelompokkan message pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram kolaborasi yang dituliskan adalah operasi atau metode yang dijalankan antara objek yang satu dengan objek yang lainnya secara keseluruhan. Oleh karena itu dapat diambil dari jalannya interaksi pada semua diagram sekuen. Untuk

menggambarkan objek dari sebuah diagram kolaborasi, dapat menggunakan pilihan Object. Dan untuk menghubungkan antar objek yang satu dengan objek yang lain digunakan Link.

Diagram kolaborasi sangat cocok untuk penggambaran interaksi sederhana antara jumlah dari benda-benda yang relatif kecil . Tetapi ada beberapa vendor menawarkan beberapa software untuk membuat dan mengedit diagram kolaborasi sehingga dapat menggambarkan interaksi jumlah objek dan pesan yang relatif besar .

Diagram kolaborasi menunjukkan informasi yang sama persis dengan diagram sekuensial, tetapi dalam bentuk dan tujuan yang berbeda. Sebagaimana diagram sekuensial, diagram kolaborasi digunakan untuk menampilkan aliran skenario tertentu di dalam use case. Jika diagram sekuensial disusun berdasarkan urutan waktu, diagram kolaborasi lebih berkonsentrasi pada hubungan antar obyek-obyek .

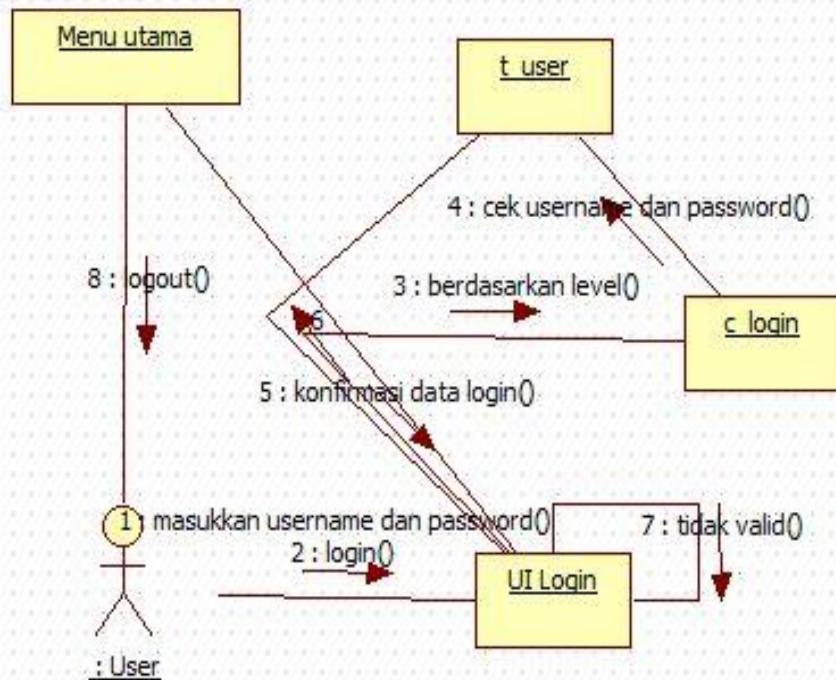
Diagram ini **menyerupai flowchart** yang menggambarkan peran, fungsi dan perilaku objek individu serta operasi keseluruhan sistem secara real time. Benda yang ditampilkan seperti persegi panjang dengan penamaan label dalam. Hubungan antara objek ditampilkan dengan garis yang menghubungkan persegi panjang. Pesan antara objek ditampilkan sebagai panah yang menghubungkan persegi panjang yang relevan dengan label yang menentukan urutan pesan.

Kita harus menempatkan obyek-obyek paling penting yang berkenaan dalam kolaborasi tersebut pada tengah-tengah dari diagram. Ini akan membantu menciptakan suatu stage / panggung / gambaran jelas dari yang secara jelas menunjukkan relasi antara obyek-obyek yang berkolaborasi.

Berikut Langkah-langkah Dasar :

- Tentukan scope / cakupan dari diagram tersebut. Sebagaimana dengan sequence diagram, cakupan dari suatu collaboration diagram dapat berperanan.
- Tempatkan obyek-obyek yang berpartisipasi dalam collaboration pada diagram. Ingatlah untuk menempatkan obyek-obyek paling penting se bisa mungkin mengarah pada tengah-tengah dari diagram.
- Bila suatu obyek tersebut memiliki properti atau menjaga suatu kondisi yang penting pada kolaborasi itu, maka tentukan lah nilai awal dari properti atau kondisi tersebut.
- Ciptakan link-link (hubungan) diantara obyek-obyek
- Ciptakan message-message yang terasosiasikan dengan tiap linknya.
- Tambahkanlah nomor urutan dari tiap message yang terkorespondensi pada urutan waktu dari message-message yang ada dalam kolaborasi tersebut.

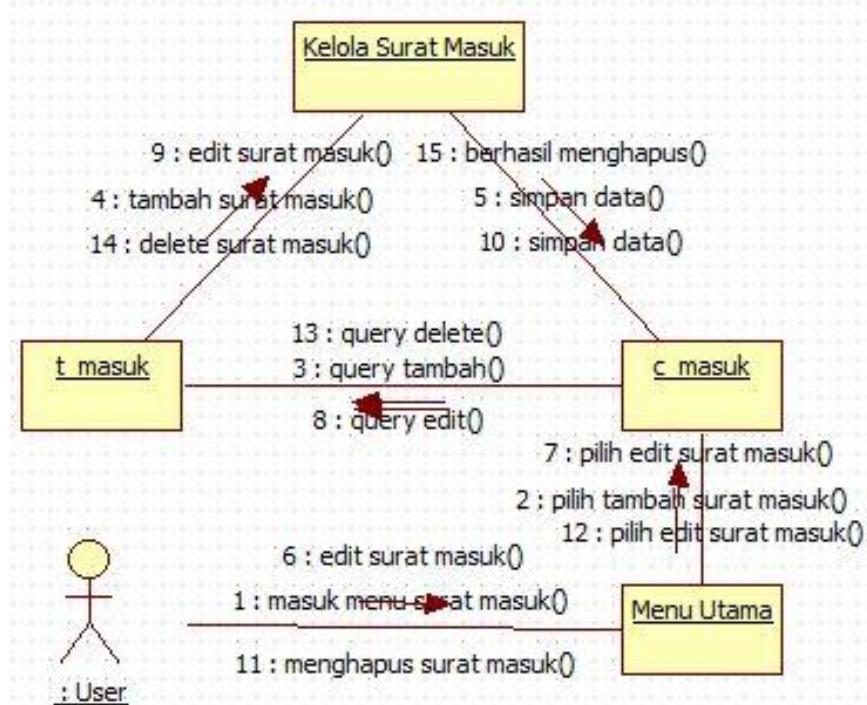
#### 5.2.4.1 Collaboration Diagram Login



Gambar 5. 11 Collaboration Diagram Login

Pada *collaboration diagram* ini menjelaskan proses *login*. *User* mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian *user* memasukkan *username* dan *password*, pada *interface login* data yang di *input* akan di cek di tabel *user*. Jika sesuai maka akan mendapatkan tampilan *form halaman utama*.

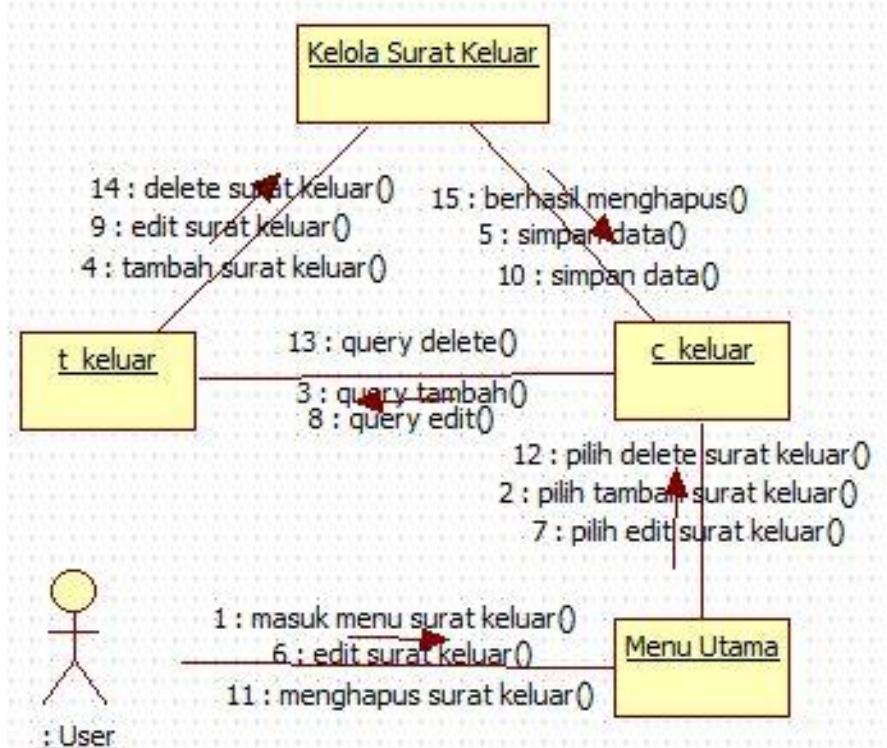
### 5.2.4.3 Collaboration Diagram Kelola Surat Masuk



Gambar 5. 12 Collaboration Diagram Kelola Surat Masuk

Pada *collaboration diagram* ini menjelaskan proses mengelola surat masuk. Bagian umum dan bidang melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu surat masuk, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *user* dapat melakukan operasi create, edit dan delete data. Selanjutnya sistem akan menyimpan perubahan tersebut.

#### 5.2.4.4 Collaboration Diagram Kelola Surat Keluar



Gambar 5. 13 Collaboration Diagram Kelola Surat Keluar

Pada *collaboration diagram* ini menjelaskan proses mengelola surat keluar. Bagian umum dan bidang melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama bagian umum atau bidang. Untuk memulai kelola ini, masuk ke menu surat keluar, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *user* dapat melakukan operasi create, edit dan delete data. Selanjutnya sistem akan menyimpan perubahan tersebut.

### **5.2.5 Activity Diagram**

Activity Diagram adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Komponen yang ada pada activity diagram antara lain :

- Activity atau state : Menunjukkan aktivitas yang dilakukan. Initial activity atau initial state : Menunjukkan awal aktivitas dimulai. Final Activity atau final state : Menunjukkan bagian akhir dari aktivitas.
- Decission : Digunakan untuk menggambarkan test kondisi untuk memastikan bahwa control flow atau object flow mengalir lebih ke satu jalur. Jumlah jalur sesuai yang diinginkan.
- Merge : Berfungsi menggabungkan flow yang dipecah oleh decission.
- Synchronization : Diabagi menjadi 2 yaitu fork dan join. Fork digunakan untuk memecah behaviour menjadi activity atau action yang paralel, sedangkan join untuk menggabungkan kembali activity atau action yang paralel.
- Swimlanes : Memecah activity diagram menjadi baris dan kolom untuk membagi tanggung jawab obyek-obyek yang melakukan aktivitas.
- Transition : Menunjukkan aktivitas selanjutnya setelah aktivitas sebelumnya. Terkadang keliru dalam penggunaan decission dengan fork. Jika decission digunakan untuk memecah aktivitas yang bersifat kondisional misal Ya-Tidak, atau user1, user2, user3, sedangkan fork untuk memecah behaviour menjadi aktivitas yang paralel seperti user dapat memilih menu hapus, tambah, edit yang dapat dilakukan secara paralel.

*Activity diagrams* adalah diagram yang menggambarkan berbagai aliran aktivitas dalam sebuah sistem yang sedang dirancang dan bagaimana

masing-masing aliran berawal, eputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir.

Activity Diagram atau Diagram aktivitas adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, atau pengulangan. Dalam *Unified Modeling Language*(UML), diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi. Selain itu diagram aktivitas juga menggambarkan alur kontrol secara garis besar.

Diagram aktivitas memiliki komponen dengan bentuk tertentu, dihubungkan dengan tanda panah. Panah tersebut mengarahkan urutan aktivitas yang terjadi, dari awal sampai akhir. Yang perlu diperhatikan yaitu diagram aktivitas bukan menggambarkan aktivitas sistem yang dilakukan aktor, tetapi menggambarkan aktivitas yang dapat dilakukan oleh sistem. .

Activity diagram pada beberapa waktu dianggap sama seperti flowchart (diagram alur), namun meskipun diagram terlihat seperti sebuah diagram alur, tetapi sebenarnya berbeda. Activity diagram menunjukkan aliran yang berbeda seperti paralel, bercabang, bersamaan dan tunggal.

### Tujuan

Tujuan utama dari Activity Diagram adalah menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses agar lebih mudah dipahami. Activity diagram juga digunakan untuk menunjukkan aliran pesan dari satu aktivitas ke aktivitas lainnya... Adapun tujuan pembuatan Activity Diagram lainnya yaitu :

- Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses
- Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis
- Membantu memahami proses secara keseluruhan
- Menggambarkan aliran paralel, bercabang dan bersamaan dari sistem

### Komponen

Komponen yang ada pada Activity Diagram antara lain :

- **Start Point/Initial State** **Mulai**  
Start Point merupakan lingkaran hitam kecil, yang menandakan tindakan awal atau titik awal aktivitas untuk setiap diagram aktivitas.



Start Point/Initial State

- **Activity(Aktivitas)**

Activity menunjukkan aktivitas yang dilakukan atau yang sedang terjadi dalam activity diagram.



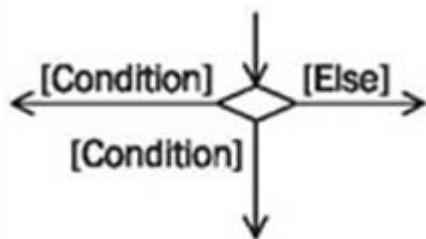
- **Action Flow(Arah)**

Action Flow digunakan untuk transisi dari suatu tindakan ke tindakan yang lain atau menunjukkan aktivitas selanjutnya setelah aktivitas sebelumnya.



- **Decision(Keputusan)**

Decision adalah suatu titik atau point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.



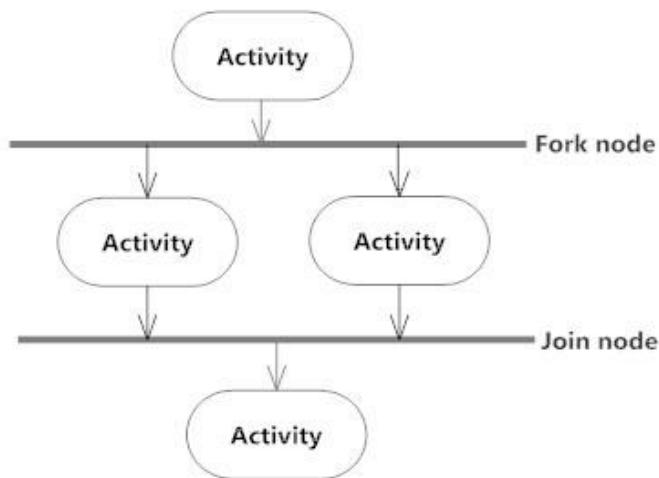
- **Synchronization**

Synchronization dibagi menjadi 2 yaitu fork dan join.

- Fork (percabangan) digunakan untuk memecah behaviour menjadi activity atau action yang paralel.

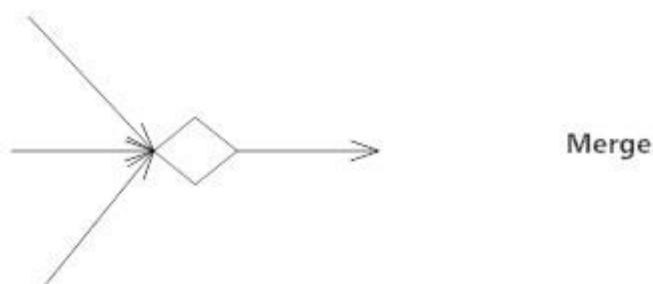
- Join (penggabungan) untuk menggabungkan kembali activity atau action yang paralel.

### Synchronization



- **Merge Event(Menggabungkan)**

Merge Event berfungsi untuk menggabungkan flow yang dipecah oleh decision.

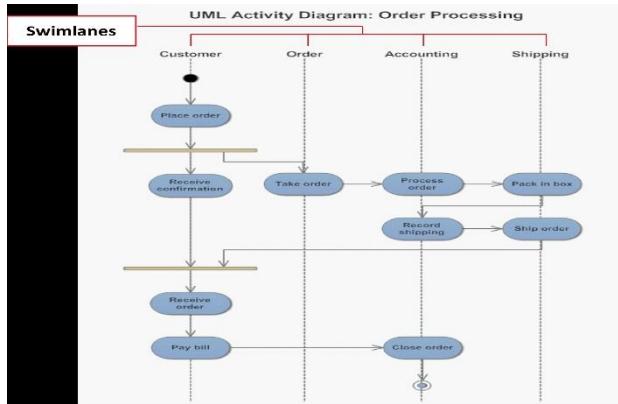


- **Swimlanes**

Swimlanes berfungsi untuk memecah activity diagram menjadi baris dan kolom untuk membagi tanggung jawab obyek-obyek yang

melakukan

aktivitas.



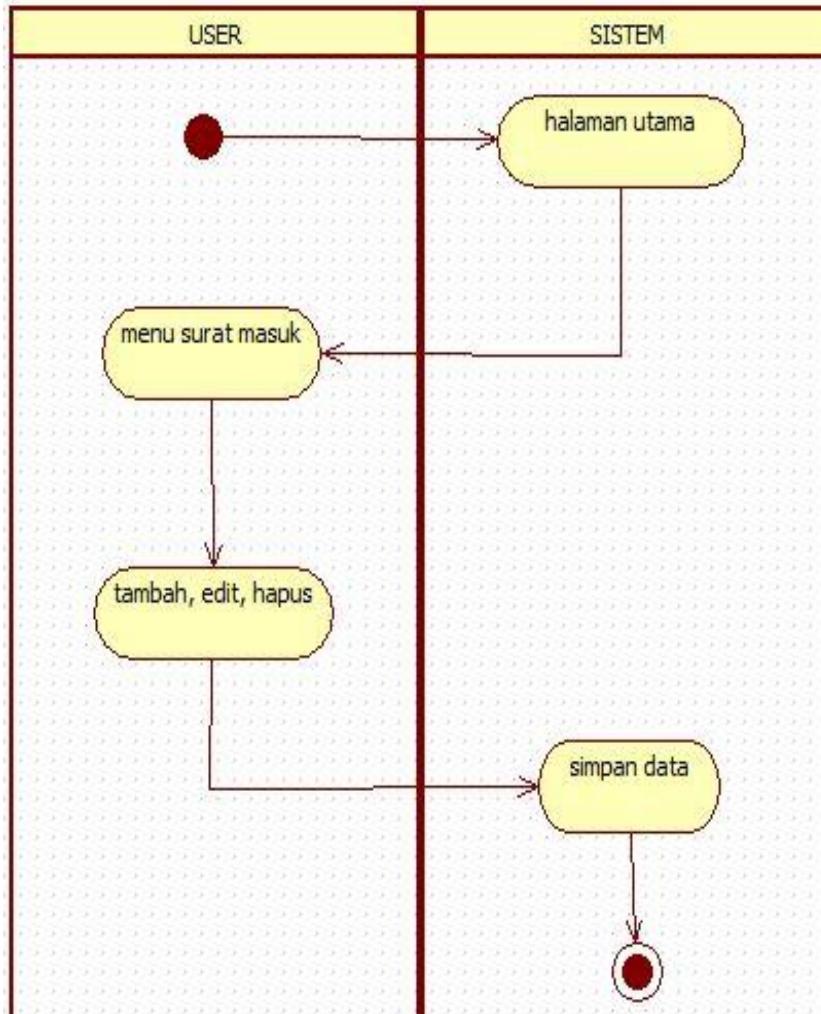
- **Final State/End Point(Titik Akhir)**

Final State menunjukkan bagian akhir dari aktivitas.



*Activity diagram* adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

#### 5.2.5.1 Activity Diagram Kelola Surat Masuk

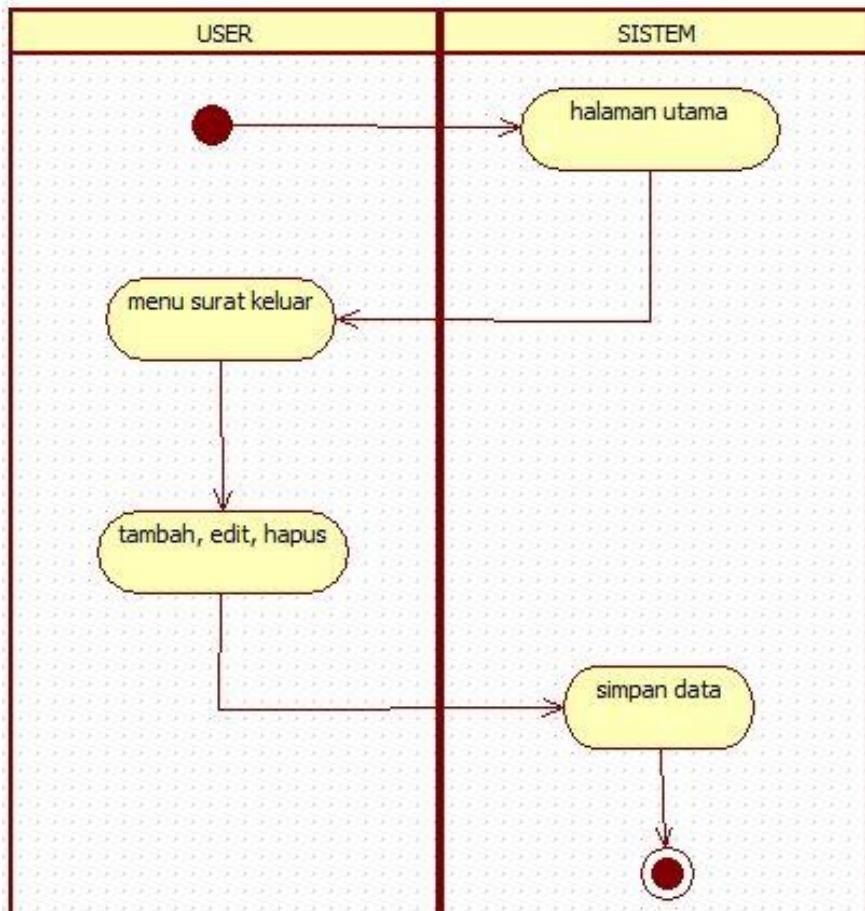


Gambar 5. 14 Activity Diagram Kelola Surat Masuk

Pada *activity diagram* ini menjelaskan proses mengelola surat masuk. Bagian umum melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu surat keluar, kemudian sistem akan melakukan koneksi *database*. Setelah *form*

tampil, *actor* dapat melakukan operasi create, edit data dan delete data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

#### 5.2.5.3 Activity Diagram Kelola Surat Keluar



Gambar 5. 15 Activity Diagram Kelola Surat Keluar

Pada *activity diagram* ini menjelaskan proses mengelola surat keluar. Bagian umum melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu surat

keluar, kemudian sistem akan melakukan koneksi *database*. Setelah *form* tampil, *actor* dapat melakukan operasi create, edit data dan delete data. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah gagal atau berhasil.

### 5.2.6 Statechart Diagram

State chart diagram adalah suatu diagram yang menggambarkan daur hidup (behavior pattern) dari sebuah objek, dari awal objek tsb diinisialisasi sampai di-destroy. Menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu obyek pada sistem sebagai akibat dari stimulans yang diterima.

Tujuan :

- Untuk memodelkan behavior/methode (lifecycle) sebuah kelas atau object
- Memperlihatkan urutan kejadian sesaat (state) yang dilalui sebuah object, transisi dari sebuah state ke state lainnya

Notasi/Simbol :

Notasi pada statechart

Notasi	Penjelasan
	State, digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu
	awal (start), digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram statechart
	Titik akhir (end), digunakan untuk menggambarkan akhir dari kejadian dalam suatu diagram statechart
[guard]	Guard, yang merupakan syarat terjadinya transisi yang bersangkutan
	Point, digunakan untuk menggambarkan apakah akan masuk (entry point) ke dalam state atau akan keluar (exit point)
<i>event</i>	Event, digunakan untuk mendeskripsikan kondisi yang menyebabkan sesuatu pada state

Tabel 6.1.

*Statechart Diagram* menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada *system* sebagai akibat dari stimulan yang diterima *statechart diagram* mendeskripsikan bagaimana suatu objek mengalami perubahan status adanya *trigger* dari *event-event*. Menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek.

**State diagram** atau *statechart diagram* menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek sehingga setiap objek memiliki sebuah diagram status.

**State diagram** adalah diagram yang digunakan untuk medeskripsikan perilaku sistem. State diagram mendeskripsikan semua kondisi yang mungkin muncul sebagai sebuah object begitu pula dengan event. Elemen-elemen didalam state diagram adalah kotak yang merepresentasikan state sebuah objek dan tanda panah untuk menunjukkan perpindahan ke state yang selanjutnya. Bagian aktivitas dari simbol state menggambarkan aktivitas object yang akan dilakukan selama didalam state.

#### Langkah-langkah untuk Menggambar State Diagram

Sebelum Anda mulai gambar, sebaiknya Anda melakukan hal dibawah ini :

- Mengidentifikasi objek penting untuk dianalisis.
- Mengidentifikasi state.
- Mengidentifikasi peristiwa.

Selanjutnya, pikirkan objek state. Misalnya, dalam e-commerce produk memiliki rilis atau tanggal yang tersedia, keadaan terjual habis, keadaan mengisi kembali, ditempatkan di state keranjang, tersimpan di daftar keinginan state, state dibeli, dan sebagainya.

Transisi tertentu tidak akan berlaku ketika suatu objek dalam keadaan tertentu, misalnya produk berada dalam keadaan dibeli atau disimpan dalam keadaan keranjang, jika keadaan sebelumnya terjual habis.

### **Dasar Statechart Simbol Diagram dan Notasi**

State merupakan situasi selama ada didalam sebuah objek. Anda dapat menggambarkan keadaannya menggunakan persegi panjang dengan sudut membulat.

- **Transisi**



Sebuah panah yang merupakan jalur antara bagian state yang berbeda dari sebuah objek. Sebuah state bisa memiliki transisi yang menunjuk kembali ke dirinya sendiri.

- **State awal**

Sebuah lingkaran diisi diikuti oleh panah mewakili keadaan awal objek.

- **Initial state**



Initial state

Panah yang menunjuk ke sebuah lingkaran.

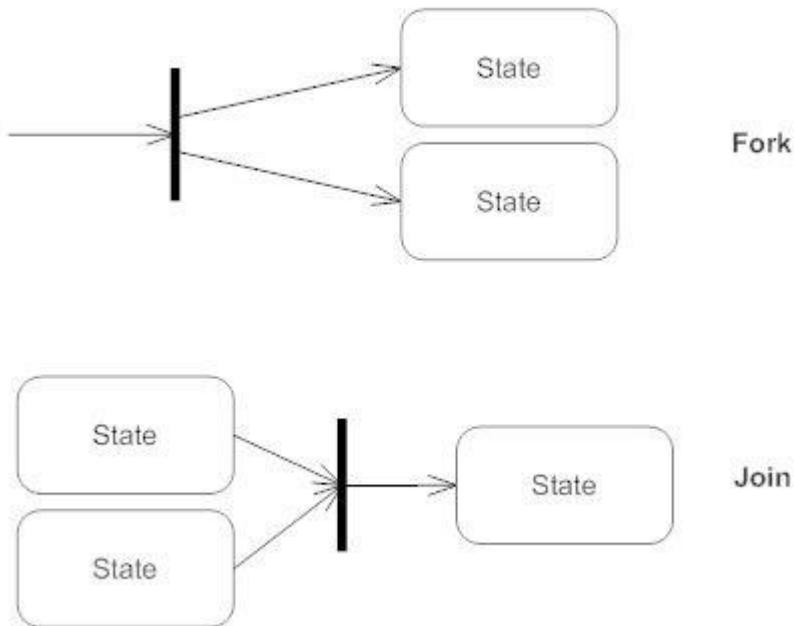
- **Final state**



Final state

Di dalam lingkaran lain mewakili keadaan akhir objek.

- **Synchronization and Splitting of Control**

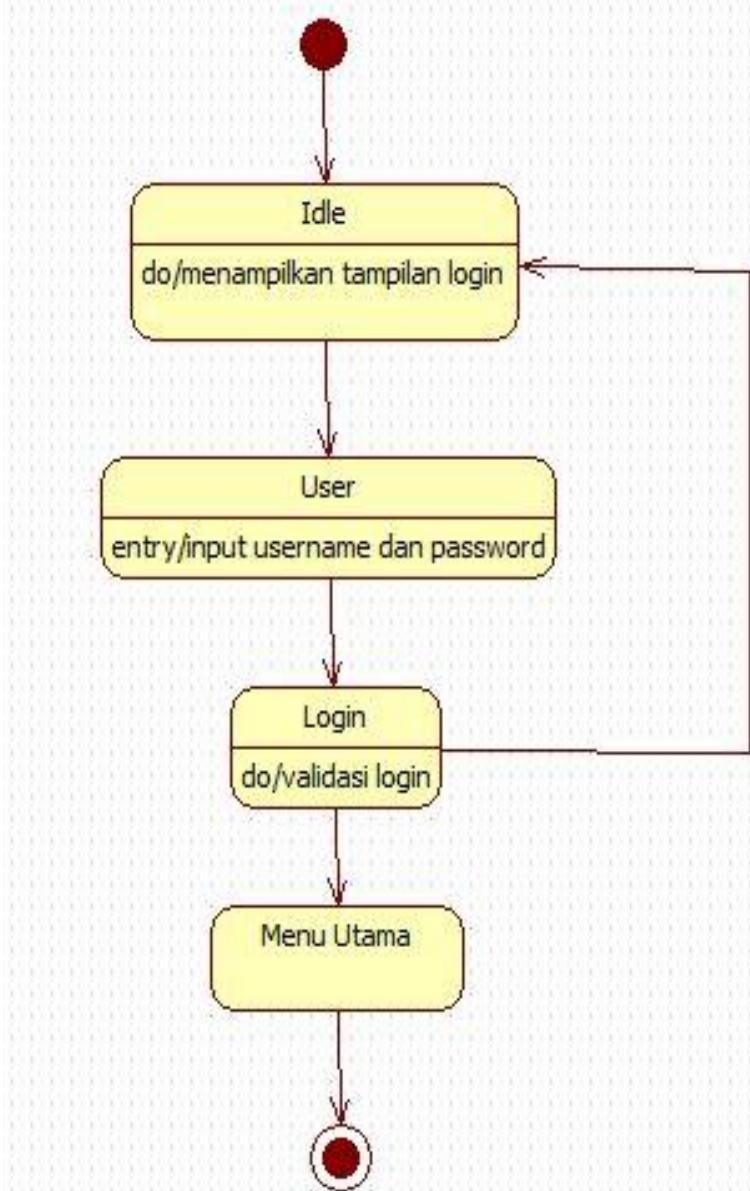


Sebuah bar pendek dengan dua transisi entering merupakan sinkronisasi kontrol. Bar pertama sering disebut fork yang dimana transisi tunggal terbagi menjadi beberapa transisi. Bar kedua disebut join, dimana transisi bersamaan mengurangi kembali ke satu.

Diagram Statechart adalah satu dari lima diagram UML yang digunakan untuk memodelkan sifat dinamik suatu sistem. Mereka mendefinisikan keadaan objek yang berbeda selama masa hidupnya dan keadaan ini diubah oleh peristiwa. Diagram statechart berguna untuk memodelkan sistem reaktif. Sistem reaktif dapat didefinisikan sebagai sistem yang merespons kejadian eksternal atau internal.

Diagram Statechart menggambarkan aliran kontrol dari satu keadaan ke keadaan lain. State didefinisikan sebagai kondisi di mana suatu benda ada dan berubah saat beberapa peristiwa dipicu. Tujuan terpenting diagram Statechart adalah memodelkan lifetime sebuah aktifitas dari awal hingga akhir.

#### 5.2.6.1 Statechart Diagram Login



Gambar 5. 16 Statechart Diagram Login

*User* masuk ke menu *login* menginputkan *username* dan *password* jika *valid* maka akan masuk ke halaman utama.

#### 5.2.6.3 Statechart Diagram Kelola Surat Masuk



Gambar 5. 17 Statechart Diagram Kelola Surat Masuk

*User* mengelola data masuk yang nanti di validasi ke *database* jika benar maka *query* berhasil.

#### 5.2.6.4 Statechart Diagram Kelola Data Surat Keluar



Gambar 5. 18 Statechart Diagram Kelola Surat Keluar

*User* mengelola data keluar yang nanti di validasi ke *database* jika benar maka *query* berhasil.

### **5.2.7 Component Diagram**

Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- Source code program perangkat lunak.
- komponen executable yang di lepas ke user.
- Basis data secara fisik.
- Sistem yang harus beradaptasi dengan sistem lain.
- Framework sistem, framework pada perangkat lunak merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi, contohnya seperti struts dari apache yang menggunakan prinsip desain Model-View-Controller (MVC) dimana soucre code program dikelompokkean berdasarkan fungsinya. dimana controller berisi source code yang menangani request dan validasi, model berisi source code yang menangani manipulasi data dan bussiness logic, dan view berisi source code yang menangani tampilan.

Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

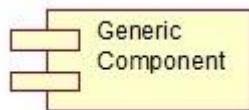
- Komponen user interface yang menangani tampilan.
- Komponen bussiness process yang menangani fungsi-fungsi proses bisnis.
- Komponen data yang menangani manipulasi data.
- Komponen security yang menangani keamanan sistem.

diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. Pada component View, akan difokuskan pada organisasi fisik system. Pertama, diputuskan bagaimana kelas-kelas akan diorganisasikan menjadi kode pustaka. Kemudia akan dilihat bagaimana perbedaan antara berkas eksekusi, berkas dynamic link library (DDL), dan berkas runtime lainnya dalam system.

### Tipe-Tipe Komponen

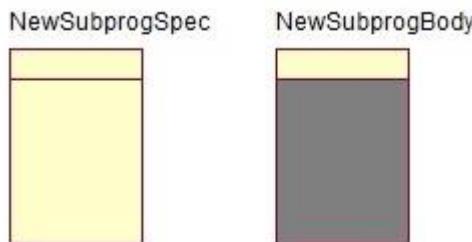
#### 1. Generic Component

komponen mempresentasikan module perangkat lunak dengan sebuah antar muka yang di Devinisikan dengan baik. Para spesifikasi komponen, kita dapat menspesifikasi tipe komponen dalam kolom stereotype( Active X, Applet, Aplikasi, DLL, Executable). Dalam UML, notasi keadaan digambarkan sebagai berikut.



## 2. Subprogram specification and Body

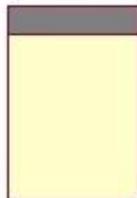
- Notasi ini mempresentasikan spesifikasi subprogram yang terlihat dan bagian implementasi. Subprogram secara tipikal adalah kumpulan beberapa subroutine. Subprogram tidak berisi devinisi kelas.



## 3. Main program

Notasi ini mempresentasikan program utama. Program utama adalah berkas yang berisi root program. Contoh, pada power builder, ada berkas yang berisi obyek Aplikasi.

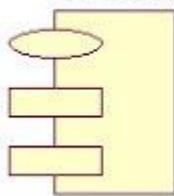
NewMainSubprog



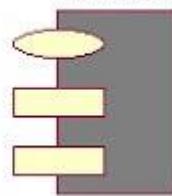
#### 4. Package Specification and Body

Sebuah paket atau package adalah implementasi kelas. Sebuah paket spesifikasi adalah berkas header, yang berisi informasi fungsi prototype untuk kelas. Di C++, spesifikasi paket adalah berkas .h. di Java, kita menggunakan notasi paket spesifikasi untuk mempresentasikan berkas .java. Sebuah package Body berisi kode untuk operasi-operasi dari kelas. Di C++, package body adalah berkas .cpp.

NewPackageSpec

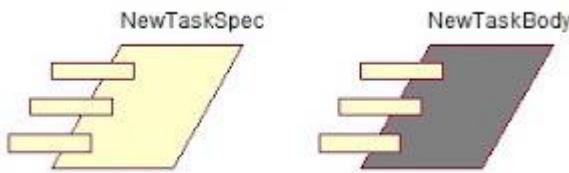


NewPackageBody



#### 5. Task Specification and Body

Notasi-notasi ini mempresentasikan paket yang memiliki Thread kontrol yang berdiri sendiri. Berkas executable biasanya mempresentasikan spesifikasi task dengan ekstensi .exe.



## 6. Database

Notasi ini mempresentasikan sebuah basis data, yang berisi satu atau beberapa skema. Pada komponen diagram, basis data ditunjukkan seperti ini



NewDatabase

## Detail komponen

### 1. Stereotype

Stereotype mengatur notasi yang akan digunakan untuk mempresentasikan komponen. Stereotype adalah (yang menggunakan notasi komponen), spesifikasi subprogram, subprogram body, program utama, paket spesifikasi, package body, executable, DLL, spesifikasi task, dan task body.

## 2. Bahasa Pemrograman

Kita dapat menandai bahasa pemrograman pada komponen-komponen dasar. Jadi, dapat dibangkitkan satu bagian model di C++, bagian lainnya di Java, bagian lainnya di Visual Basic dan sebagainya.

Rose Enterprise berisi tambahan untuk ANSI C++, Ada 83, Ada 95, CORBA, C++, Java, Visual basic, Visual C++, Web modeler, XML/DTD, Dan Oracle 8. Ada banyak lagi tambahan tersedia dari beberapa vendor untuk memperluas kemampuan Rose. Untuk bahasa pemrograman lain (Power Builder, Forte, Visual Age for Java, etc) mungkin akan dipesan juga.

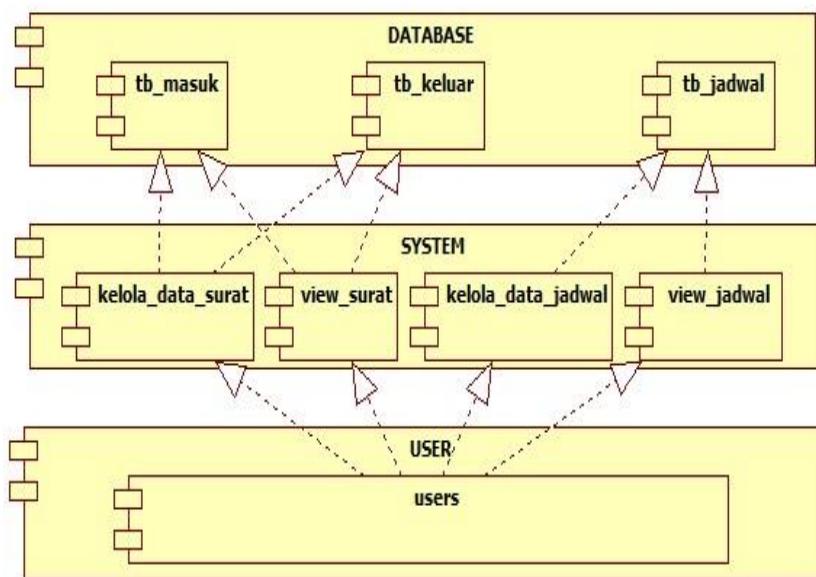
## 3. Deklarasi

Ada bagian untuk mencantumkan deklarasi pelengkap yang ditambahkan pada saat generate code untuk setiap component. Deklarasi termasuk bahasa pemrograman pernyataan spesifik yang digunakan untuk mendeklarasi variable, kelas-kelas, dan sebagainya. Pernyataan #include kepada C++ juga termasuk deklarasi ini.

## 4. Kelas-kelas

Sebelum men-generate code untuk sebuah kelas, maka harus dipetakan dulu kekomponen. Pemetaan ini membantu Rose mengetahui kelas-kelas mana saja yang akan dipetakan dalam berkas fisik kode.

*Component diagram* yaitu salah satu jenis *diagram* pada *UML* yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa *file* data atau *.exe*, *source code*, *table*, dokumen, dll.



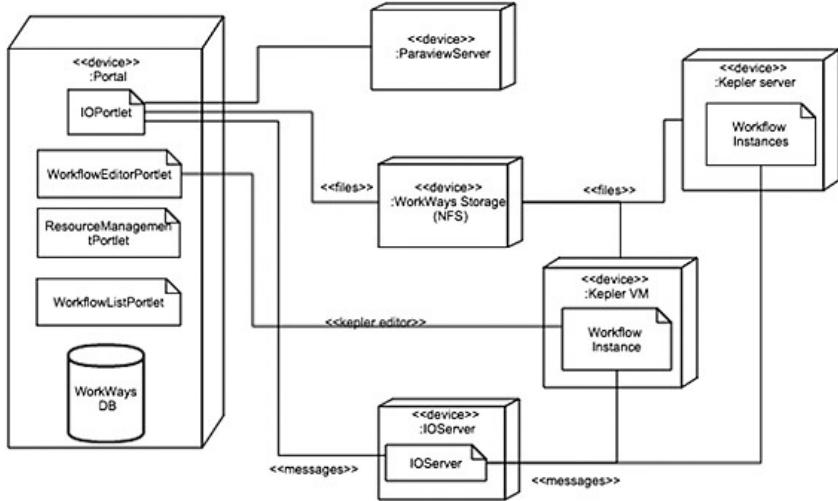
Gambar 5. 19 Component Diagram

### **5.2.8 Deployment Diagram**

**Deployment diagram** adalah diagram yang digunakan memetakan software ke processing node. Menunjukkan konfigurasi elemen pemroses pada saat run time dan software yang ada di dalamnya.

**Deployment diagram** adalah salah satu model diagram dalam UML untuk mengerahkan artifact dalam node. Deployment diagram digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis Object Oriented yang akan dibangun. Tujuan atau fungsi dari deployment diagram yaitu untuk menggambarkan/memvisualisasikan secara umum proses yang terjadi pada suatu sistem/software.

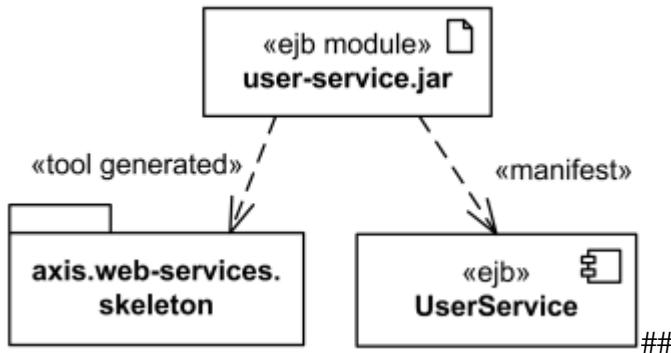
Node dalam UML merepresentasikan hardware atau software execution environments. Node bisa terhubung melalui communication path untuk membuat system jaringan dari arbitrary complexity. Artifacts dalam UML mempresentasikan Spesifikasi dari bentuk physic informasi yang digunakan atau dihasilkan development process.



Ada beberapa hal yang sering digambarkan dalam *deployment diagram* yaitu:

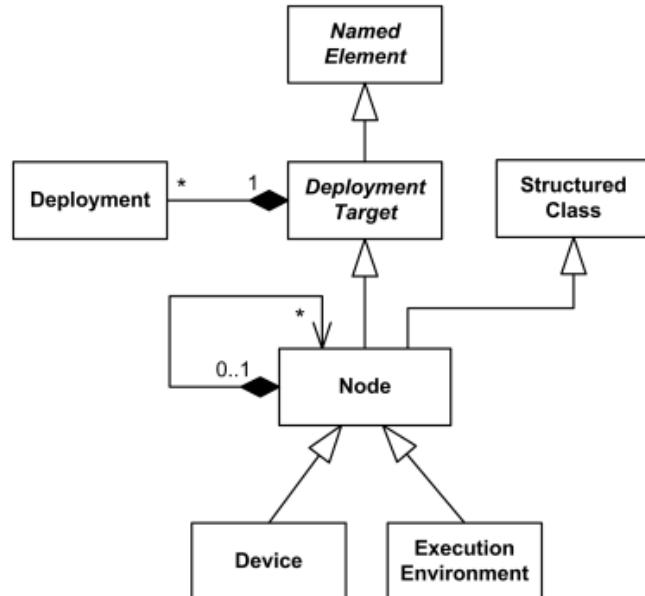
### 1. Manifestation

Manifestation adalah hubungan abstraction yang mempresentasikan physical rendering atau hasil implementasi satu atau lebih model element dari artifact. Atau pemanfaatan model element dari konstruksi artifact. Artifact memanifest atau menunjukkan satu atau lebih dari model element. Artifact bisa memiliki satu atau lebih manifestation. Setiap darinya mempresentasikan packageable element.



## 2. Deployment target

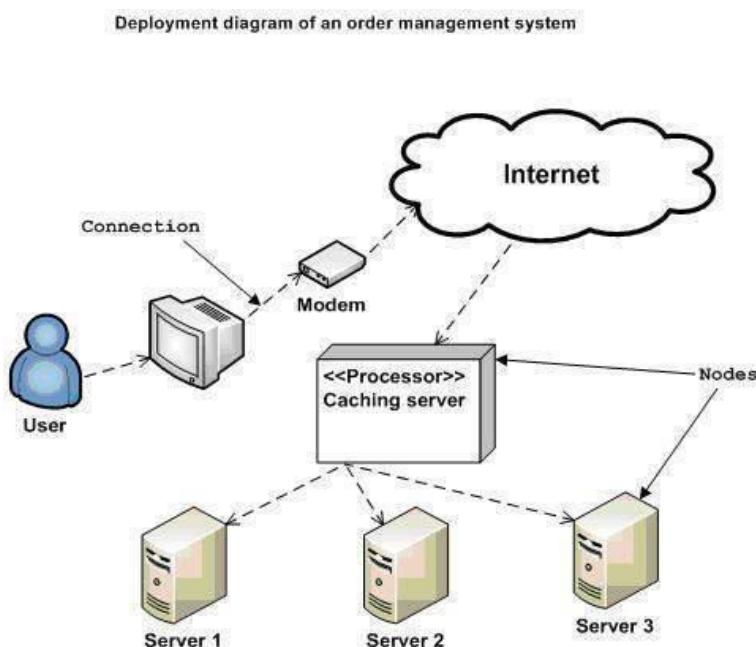
Artifacts dikerahkan ke deployment targets. Deployment target adalah lokasi untuk artifact yang telah dikerahkan. Deployment target tidak memiliki notation sendiri, tapi melihat notations untuk subclasses.

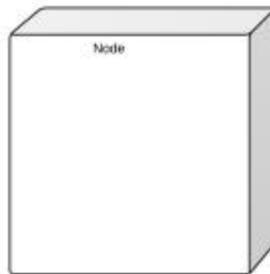


### 3. Node

Node adalah deployment target yang merepresentasikan sumber daya komputasi dimana artifact bisa menggunakan untuk mengeksekusi. Node diasosiasikan dengan deployment dari artifacts dan dapat diasosiasikan secara tidak langsung dengan packageable elements yang melibatkan ke dalam manifestation dari artifact yang dikerahkan ke node. Node dapat saling berhubungan dengan communication path.

Node dapat digambarkan dengan bentuk kubus tiga dimensi. Node juga dapat digambarkan dalam bentuk visual atau gabungan antara node dan visual



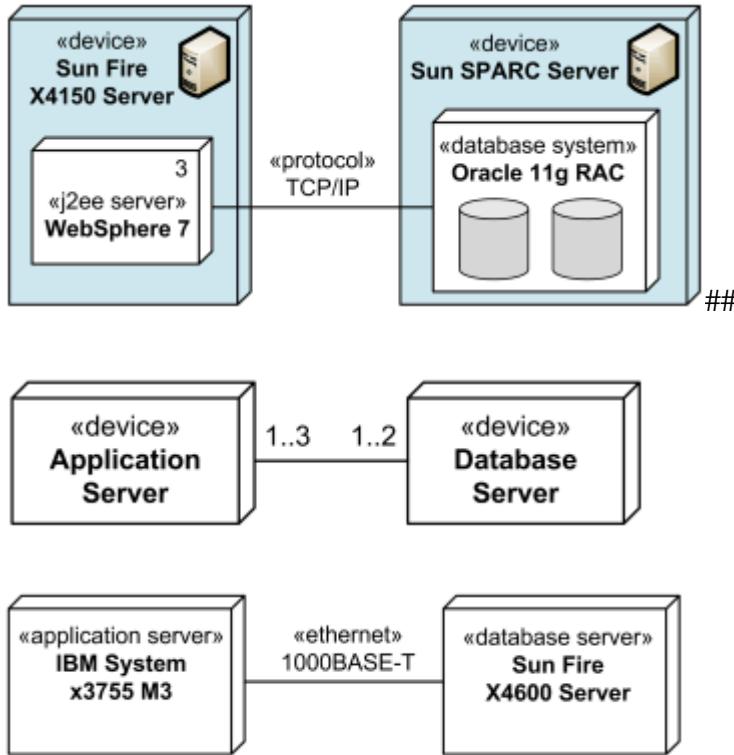


Node dibagi menjadi dua yaitu device dan execution environment.

- Device merepresentasikan hardware seperti : pc, laptop, handphone, dll
- Execution environment merepresentasikan software containers seperti : OS, JVM, application servers, portal servers, dll

#### **4. Communication Path**

Communication path adalah asosiasi antara dua deployment target, melalui pertukaran sinyal dan pesan. Communication path digambarkan dengan bentuk garis utuh yang menghubungkan antara dua node. Jenis komunikasi yang dimodelkan dapat ditampilkan menggunakan stereotip yang tepat. Saat deployment targetnya adalah execution environments. Communication path akan merepresentasikan beberapa protocol



## 5. Deployment

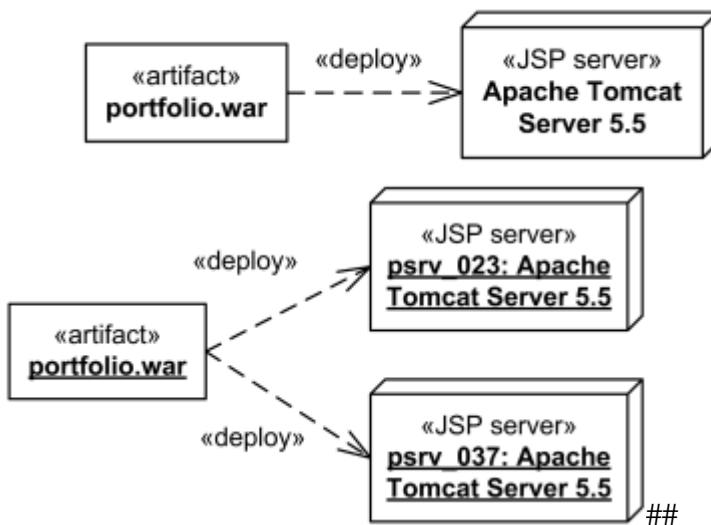
*Deployment* adalah hubungan dependency yang mendeskripsikan alokasi dari artifact ke deployment target. Deployment juga dapat didefinisikan instansi level sebagai alokasi spesifik instansi artifact menuju ke spesifik instansi dari deployment target

*Component deployment* adalah deployment dari satu atau lebih artifact atau instansi artifact, bersifat opsional yang terparameterisasi oleh *deployment specification*

*Deployment* bisa diperlihatkan melalui tiga cara yaitu :

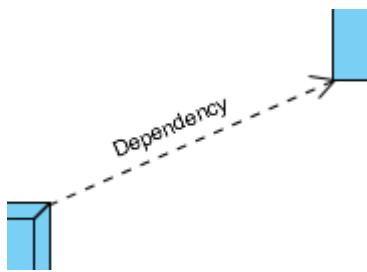
- Dengan *deployed artifact* yang terkandung di deployment target,
- Dengan bentuk tulisan daftar deployed artifact dalam deployment target
- Dengan bentuk persegi panjang dengan nama deployment yang terletak dipojok kiri atas.

Kata *deployment* di *heading* disingkat menjadi *dep*



## 6. Dependency

*Dependency* adalah hubungan yang menandakan bahwa satu atau sekumpulan model element membutuhkan model element lainnya untuk spesifikasi atau implementasinya. *Dependency* biasa juga disebut supplier yang menyediakan sesuatu untuk client. Memodifikasi supplier dapat mempengaruhi client element



## 7. Deployment Specification

*Deployment specification* adalah artifact yang menyespesifikasikan set dari deployment properties yang menentukan parameter – parameter eksekusi dari component artifact yang dikerahkan ke node.

*Deployment specification* dapat ditujukan spesifik tipe container untuk componenet – component.

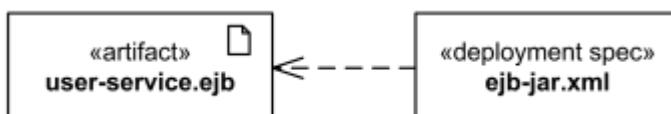
*Deployment specification* adalah mekanisme umum untuk memparameteri deployment relationship.

*Deployment specification* di *specification level* dirender sebagai *classifier* dalam kotak dengan opsi deployment properties di dalam compartment. Artifact yang mengimplementasikan deployment specification properties di level instansi adalah deployment descriptor. deployment descriptor ditandai dengan garis bawah di kotak classifier

«deployment spec» <u>ejb-jar.xml</u>	«deployment spec» <u>ejb-jar.xml</u>
ejb-name: UserService session-type: Stateful transaction-type: Bean	ejb-name: String session-type: String transaction-type: String

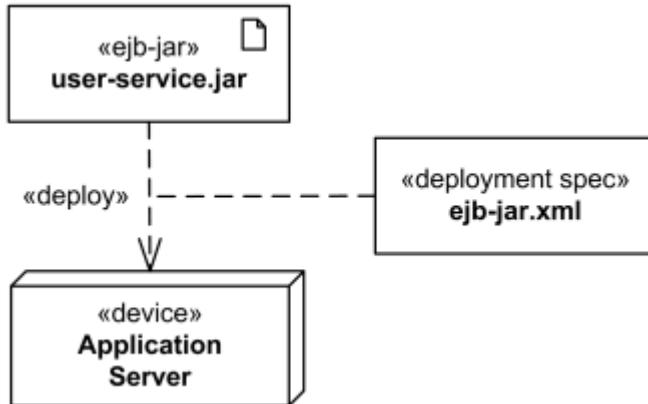
## 8. Deployment Specification Dependency

*A deployment specification* bisa ditampilkan sebagai kotak classifier yang melampirkan component artifact menggunakan panah dependency yang menunjuk ke *deployed artifact*.



## 9. Deployment Specification Association

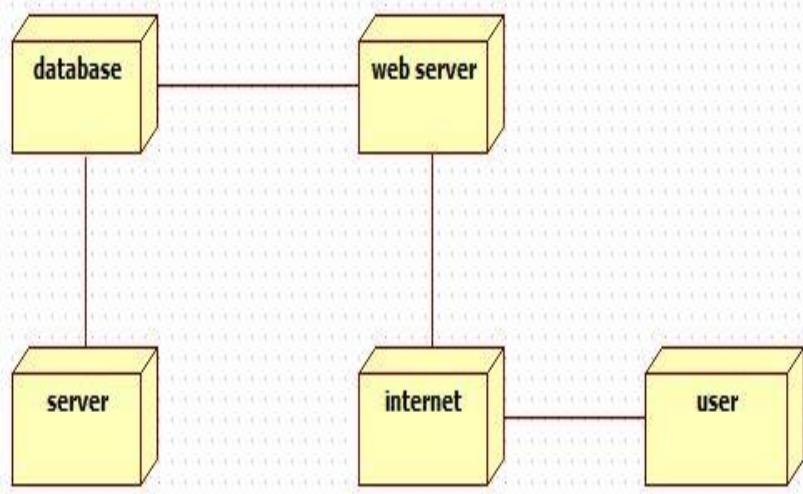
*Deployment specification* bisa diasosiasikan dengan deployment dari sebuah component artifact dalam sebuah node. Dalam kasus ini deployment specification bisa ditunjukan sebagai kotak classifie yang dilampirkan ke deployment.



## 10. Component

*Component* merepresentasikan bagian modular dari sebuah sistem yang mengkapsulasi isi dan yang memanifest dapat diganti dalam lingkungannya. component mendefinisikan perilaku dalam hal interface yang disediakan dan diperlukan. component dimodelkan sepanjang siklus hidup pengembangan dan berturut-turut disempurnakan menjadi deployment dan run – time. deployment specification dapat mendefinisikan nilai-nilai yang parameterisasi eksekusi komponen.

*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi



*Gambar 5. 20 Deployment Diagram*

### **5.2.9 Perancangan Antar Muka**

User interface adalah bagian visual dari website, aplikasi software atau device hardware yang memastikan bagaimana seorang user berinteraksi dengan aplikasi atau website tersebut serta bagaimana informasi ditampilkan di layarnya. User interface sendiri menggabungkan konsep desain visual, desain interasi, dan infrastruktur informasi. Tujuan dari user interface adalah untuk meningkatkan usability dan tentunya user experience.

User interface adalah salah satu faktor yang menentukan peningkatan traffic website Anda. UI bisa menjadi faktor apakah seseorang tertarik untuk mengunjungi dan meng-explore website Anda. Kalau mereka senang dengan tampilan interface Anda dan experience mereka menggunakan website Anda, maka kemungkinan besar mereka akan mengunjungi website Anda kembali. Jadi Anda harus selalu memperhatikan UI website Anda saat Anda mendesain website.

Satu hal yang mungkin jarang diketahui orang banyak adalah seberapa pentingnya desain pada user interface. Bahkan kadang bentuk sebuah tombol saja bisa menentukan apakah seseorang mengerti bagaimana menavigasi website atau aplikasi tersebut? Desain user interface memang sangat penting karena itu akan menentukan bagaimana seseorang berinteraksi dengan website dan aplikasi tersebut. User interface juga menentukan apakah pengunjung website atau pengguna aplikasi dapat menavigasi website atau aplikasi dengan mudah.

Saat seseorang mengunjungi website Anda, user experience harus menjadi prioritas Anda. Kalau orang memiliki user experience yang baik di website Anda, Anda akan mendapat conversion rate yang lebih tinggi. Selain itu, mereka juga akan mempromosikan website atau aplikasi Anda ke orang-orang lain. Kalau mereka mengalami pengalaman buruk dengan website atau aplikasi Anda, pasti mereka tidak mau menggunakannya lagi. Jika Anda adalah bisnis online, ini tentunya penting. Untuk memiliki user experience yang baik, Anda memerlukan user interface yang mudah dinavigasi dan digunakan.

User interface menentukan customer base Anda. Selama user interface Anda mudah dimengerti, user experience pengunjung website atau pengguna aplikasi juga akan tetap baik. Dengan begitu, kemungkinan mereka untuk tetap datang ke website Anda dan menggunakan website atau aplikasi Anda juga lebih tinggi.

Di luar sana ada berbagai informasi yang menyarankan teknik dan pola desain interface yang dianggap baik dan benar. Memang kalau Anda mengikuti saran-saran tersebut, kemungkinan besar Anda akan membuat website atau aplikasi dengan UI yang baik. Tetapi mungkin Anda juga perlu checklist karakteristik apa saja yang sebenarnya membuat sebuah UI dibilang bagus.

### **Jelas**

Memiliki UI yang jelas adalah salah satu elemen penting dalam desain user interface. Tentunya tujuan dari desain UI adalah agar orang-orang bisa menggunakan dan berinteraksi dengan sistem Anda dengan mudah. Jika orang-orang tidak bisa mengerti bagaimana cara menggunakan dan menavigasi website Anda mereka pasti akan bingung.

## **Singkat**

Tentunya memiliki UI yang jelas akan membuat UX Anda bagus, tetapi Anda juga harus hati-hati agar penjelasan Anda tidak terlalu panjang. Kalau Anda meletakkan definisi dan penjelasan pada setiap bagian website Anda, website Anda akan terlihat lebih berantakan. Kemungkinan besar dengan bertumbuhnya website Anda, interface Anda juga akan terus berkembang. Kalau Anda meletakkan terlalu banyak penjelasan, user Anda nantinya akan menghabiskan waktu membaca penjelasannya.

UI boleh jelas, tapi juga harus singkat. Kalau memang diperlukan penjelasan, usahakan agar Anda bisa menjelaskannya dalam satu kalimat. Jika Anda bisa memberi label hanya dengan satu kata itu tentunya akan lebih baik. Jangan buang waktu user Anda. Mungkin memastikan bahwa semuanya singkat tapi jelas bisa menjadi tantangan. Tetapi kalau Anda bisa melakukan itu, UI website Anda akan menjadi memuaskan.

## **Familiar**

Mungkin salah satu kata yang sering Anda dengar saat membahas desain web adalah intuitif. Apa sebenarnya arti dari kata itu? Secara simple, intuitif artinya layout Anda bisa dimengerti dengan mudah secara natural dan hanya dengan menggunakan instinc. Tetapi bagaimana cara membuat layout website Anda intuitif? Anda perlu menggunakan desain web yang familiar.

Familiar yang dimaksud disini adalah sesuatu yang sudah pernah dilihat sebelumnya. Kalau Anda familiar dengan sesuatu, Anda pasti tahu kan apa yang harus dilakukan? Coba pikirkan hal-hal apa yang sudah pasti familiar bagi user website Anda dan masukkan hal-hal ini ke desain web Anda.

## **Responsif**

Bagi Anda yang mempelajari web design dan SEO mungkin sudah familiar dengan istilah responsive. Untuk UI sendiri, responsive memiliki beberapa arti yang berbeda. Pertama, responsive berarti cepat. Interface website Anda harus bisa bekerja dengan cepat. Kalau Anda perlu menunggu sebuah website loading lama pasti Anda juga malas kan. Jika interface Anda bisa loading dengan cepat tentunya user experience Anda juga akan semakin baik.

Arti lain dari responsive pada UI juga harus bisa memberitahu user apa yang sedang terjadi di halaman itu. Misalnya, jika seorang user mengklik suatu tombol di website Anda, apakah mereka sudah berhasil

menekan tombol tersebut? Mungkin text pada tombol Anda bisa menjadi kata “loading” jika mereka sudah berhasil menekan tombolnya. Bisa juga Anda menggunakan progress bar seperti loading Gmail sebagai indikator kalau website sedang loading.

## **Konsisten**

Dalam mengembangkan user interface, konsistensi pada interface dapat membantu user untuk mengerti pola. Dari satu interface, mereka bisa mempelajari apa kegunaan tombol, tabs, icons, dan berbagai elemen yang ada pada interface tersebut. Jika mereka nantinya menemukan interface yang mirip, mereka bisa mengerti apa kegunaan elemen-elemen yang ada pada interface tersebut. Dengan begitu, mereka bisa mengerjakan sesuatu dengan lebih cepat dan mempelajari fitur-fitur baru dengan lebih cepat.

Salah satu perusahaan yang kerap melakukan ini adalah Microsoft dengan program Microsoft Officenya. Jika Anda perhatikan dari dulu, meskipun mereka selalu mengupdate program-program Microsoft Office nya seperti Word, Excel, dan Power Point, elemen-elemen yang ada pada ketiga program tersebut tetap konsisten. Meskipun interfacenya berubah, Anda tahu bahwa untuk menyimpan file, Anda bisa klik pada icon disket dan untuk menge-print Anda tinggal klik pada icon printer.

## **Menarik**

Poin yang satu ini mungkin bisa dibilang subjektif tetapi interface akan lebih baik jika mereka memiliki tampilan menarik. Yang dimaksud dengan menarik disini adalah interface Anda menarik untuk digunakan. Memang kalau Anda bisa membuat UI Anda simple, mudah digunakan, efisien, dan responsive, Anda sudah memiliki UI yang baik. Tetapi kalau Anda juga bisa membuatnya menarik tentunya akan lebih asik untuk digunakan, bukan? Anda juga akan membuat customer Anda lebih senang menggunakan website Anda.

Tentunya apa yang dianggap menarik untuk website atau aplikasi Anda juga perlu disesuaikan dengan pasar dan audience Anda. Jadi tampilan Anda harus menarik sesuai dengan audience Anda. Tetapi Anda juga harus tetap memerhatikan fungsi website Anda agar mereka tetap berfungsi dengan baik.

## **Efisien**

User interface yang baik harus memastikan bahwa website dan aplikasinya bisa digunakan dengan efisien. Agar Anda bisa membuat UI yang efisien Anda perlu tahu dulu apa yang ingin user capai dan biarkan mereka melakukan langkah-langkahnya tanpa banyak masalah. Anda perlu mengidentifikasi bagaimana website atau aplikasi Anda bekerja. Apa saja fungsinya dan apa kegunaannya. Buat interface yang memudahkan user untuk mencapai tujuan mereka.

## **Forgiving**

Anda mungkin pernah melakukan kesalahan saat sedang mengakses website. Misalnya, Anda salah menghapus informasi, UI yang baik akan membantu Anda mengembalikannya dengan cepat. Selain itu, kalau misalnya user Anda masuk ke halaman website yang error, apakah Anda menyarankan mereka untuk mencoba mengarah ke halaman lain website? Tanpa Anda sadari, hal-hal ini bisa membantu meningkatkan user experience Anda.

### **5.2.9.1 User Interface Login User**



*Gambar 5. 21 User Interface Login*

Keterangan Gambar :

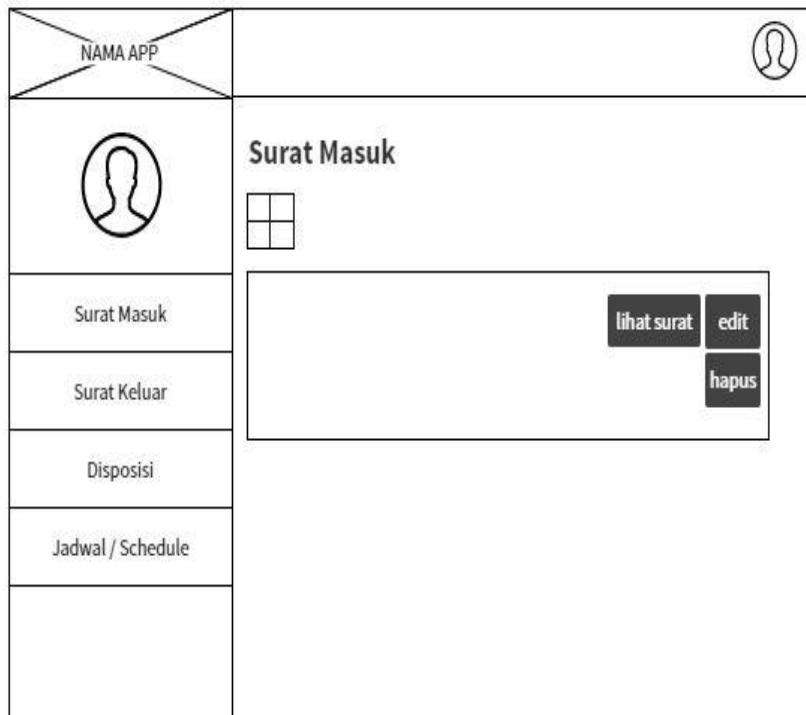
*Textbox 1* : Input Username

*Textbox 2* : Input Password

*Button 1* : Tombol Login

*Label 1* : Dinas Perhubungan Provinsi Jawa Barat

#### **5.2.9.2 User Interface Halaman Surat Masuk**



*Gambar 5. 22 User Interface Halaman Surat Masuk*

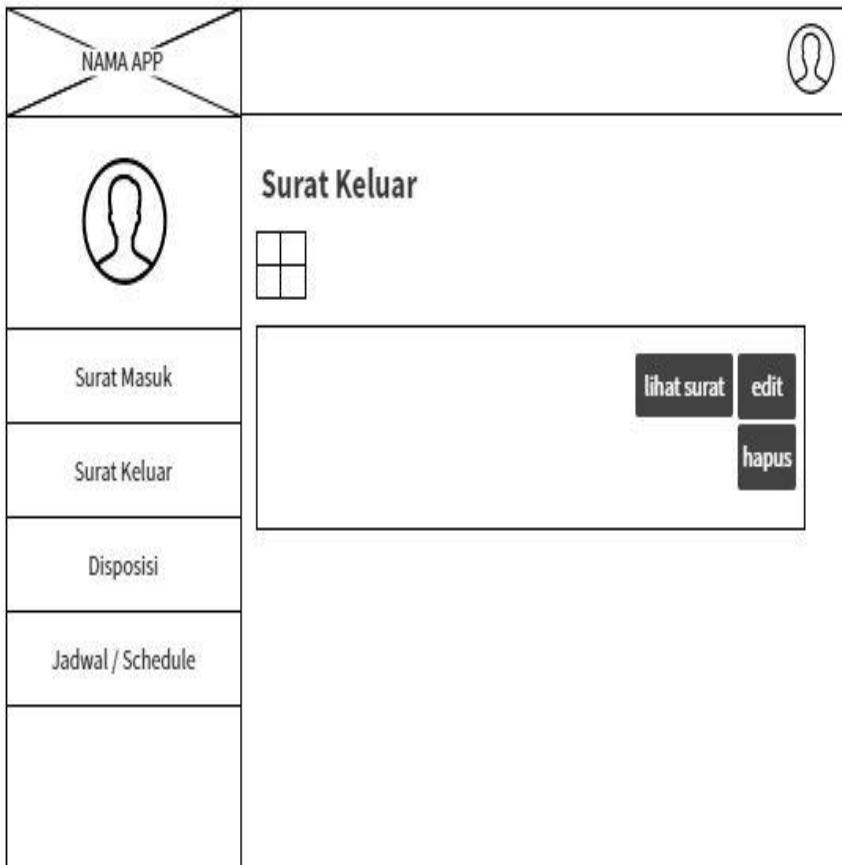
Keterangan Gambar :

*Label 1* : Surat Masuk  
*Label 2* : Surat Keluar  
*Label 3* : Disposisi  
*Label 4* : Jadwal / Schedule  
*Button* : Tambah, Edit, Hapus, Lihat Surat

*Profile Picture*

*Logo*

### **5.2.9.3 User Interface Halaman Surat Keluar**



*Gambar 5. 23 User Interface Halaman Surat Keluar*

Keterangan Gambar :

*Label 1* : Surat Masuk

*Label 2* : Surat Keluar

*Label 3* : Disposisi

*Label 4* : Jadwal / Schedule

*Button* : Tambah, Edit, Hapus, Lihat Surat

*Profile Picture*

*Logo*

#### **5.2.9.5 User Interface Halaman Tambah Surat Masuk**

The diagram illustrates the user interface for the 'Tambah Surat Masuk' (Add Incoming Letter) page. On the left, there is a vertical sidebar with several menu items: 'NAMA APP' (app name), a profile picture placeholder, 'Surat Masuk' (Incoming Letter), 'Surat Keluar' (Outgoing Letter), 'Disposisi' (Disposition), 'Jadwal / Schedule', and an empty space at the bottom. The main content area is titled 'Tambah Surat Masuk'. It contains input fields for the date ('12 May 2016') and time ('00:00'), followed by five empty text input boxes for letter content. At the bottom of the main area is a dark grey button labeled 'Tambah' (Add).

*Gambar 5. 24 User Interface Halaman Tambah Surat Masuk*

Keterangan Gambar :

*Textbox 1* : Nomor Surat

*Datepicker* : Tanggal Surat

*Textbox 2* : Sifat Surat

*Textbox 3* : Isi Ringkas

*Textbox 4* : Surat Dari

*Textbox 5* : Kepada

*Textbox 6* : Diteruskan

*Textbox 7* : Pengolah

*Button* : Tambah

#### **5.2.9.6 User Interface Halaman Tambah Surat Keluar**

NAMA APP	
	<b>Tambah Surat Masuk</b>
Surat Masuk	<input type="text"/> 12 May 2016 <input type="button" value="Calendar"/>
Surat Keluar	<input type="text"/>
Disposisi	<input type="text"/>
Jadwal / Schedule	<input type="text"/>
	<b>Tambah</b>

*Gambar 5. 25 User Interface Halaman Tambah Data User*

Keterangan Gambar :

*Textbox 1* : Nomor Surat

*Datepicker* : Tanggal Surat

*Textbox 2* : Sifat Surat

*Textbox 3* : Prihal

*Textbox 4* : Surat Dari

*Textbox 5* : Kepada

*Button* : Tambah

## **BAB VI**

---

### **6.1 PENGUJIAN DENGAN BLACKBOX**

#### **6.1.1 PENGENALAN BLACKBOX**

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu koatak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya(interface nya) , fungsionalitasnya.tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui input dan output).

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian white-box). pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan.Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Metode ujicoba blackbox memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba blackbox memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba blackbox bukan merupakan alternatif dari ujicoba whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode whitebox.

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

Metode ujicoba blackbox memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba blackbox memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba blackbox bukan merupakan alternatif dari ujicoba whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode whitebox.

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal

4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

Tidak seperti metode whitebox yang dilaksanakan diawal proses, ujicoba blackbox diaplikasikan dibeberapa tahapan berikutnya. Karena ujicoba blackbox dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi domain. Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji?
2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan-batasan kelas data diisolasi?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan ujicoba blackbox, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

- kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari ujikasus tambahan harus didesain untuk mencapai ujicoba yang cukup beralasan
- Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, daripada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik Equivalence Partitioning Equivalence partitioning merupakan metode ujicoba blackbox yang membagi domain input dari program menjadi beberapa kelas data dari kasus ujicoba yang dihasilkan. Kasus uji penanganan single yang ideal menemukan sejumlah

kesalahan (misalnya : kesalahan pemrosesan dari seluruh data karakter) yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati.

Equivalence partitioning berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk Equivalence partitioning berdasarkan pada evaluasi dari ekuivalensi jenis/class untuk kondisi input. Class-class yang ekuivalen merepresentasikan sekumpulan keadaan valid dan invalid untuk kondisi input. Biasanya kondisi input dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi boolean.

Ekuivalensi class dapat didefinisikan dengan panduan berikut :

1. Jika kondisi input menspesifikasikan kisaran/range, maka didefinisikan 1 yang valid dan 2 yang invalid untuk equivalence class
2. Jika kondisi input memerlukan nilai yang spesifik, maka didefinisikan 1 yang valid dan 2 yang invalid untuk equivalence class
3. Jika kondisi input menspesifikasikan anggota dari himpunan, maka didefinisikan 1 yang valid dan 1 yang invalid untuk equivalence class
4. Jika kondisi input adalah boolean, maka didefinisikan 1 yang valid dan 1 yang invalid untuk equivalence class

Misalkan, terdapat data terpelihara untuk sebuah aplikasi perbankan otomatis. User dapat mengaksesnya dari komputer pribadinya dengan menyediakan password 6 digit, dan mengikuti serangkaian perintah keyword yang mengakses berbagai fungsi perbankan. Software yang digunakan untuk aplikasi perbankan menerima data dalam bentuk :

- Area code – blank atau 3 digit nomor
- Prefix – 3 digit nomor yang tidak diawali oleh 0 atau 1
- Suffix – 4 digit nomor
- Password – 6 digit alphanumerik
- Commands – ”check”, ”deposit”, ”bill pay”, dsb

Kondisi input yang dihubungkan dengan setiap elemen data untuk aplikasi perbankan dapat dispesifikasikan sebagai :

1. Area code : kondisi input, Boolean – area code boleh ada maupun tidak
- Kondisi input, Range – nilai didefinisikan antara 200 dan 999, dengan beberapa pengecualian khusus (misal : tidak ada nilai > 905) dan syarat (misal : seluruh area code memiliki angka 0 atau 1 pada posisi digit ke-2)
2. Prefix : kondisi input, Range – nilai yang dispesifikasikan > 200
3. Suffix : kondisi input, Value – sepanjang 4 digit
4. Password : kondisi input, Boolean – Password boleh ada maupun tidak
- kondisi input, Value – 6 string karakter
5. Command : kondisi input, Set – mengandung perintah-perintah yang ada diatas

Aplikasikan panduan untuk derivasi dari class-class yang ekuivalen, kasus uji untuk setiap domain input data item dapat di bentuk dan dieksekusi. Kasus uji dipilih sehingga sejumlah atribut dari equivalence class dieksekusi sekali saja.

Black Box pengujian adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian white-box). Pengetahuan khusus dari

kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Pengujian pada Black Box berusaha menemukan kesalahan seperti:

- Fungsi-fungsi yang tidak benar atau hilang
- Kesalahan interface
- Kesalahan dalam struktur data atau akses database eksternal
- Kesalahan kinerja
- Inisialisasi dan kesalahan terminasi

Teknik khas Black Box Testing desain meliputi:

#### *1. DECISION TABLE*

Decision Table adalah cara yang tepat belum kompak untuk model logika rumit, seperti diagram alur dan jika-then-else dan switch-laporan kasus, kondisi mengaitkan dengan tindakan untuk melakukan, tetapi dalam banyak kasus melakukannya dengan cara yang lebih elegan.

Pada tahun 1960-an dan 1970-an berbagai “Decision Table Based“ bahasa seperti Filetab sangat populer untuk pemrograman bisnis.

## 2. ALL-PAIRS TESTING

All-pairs testing atau pairwise testing adalah metode pengujian perangkat lunak kombinatorial bahwa, untuk setiap pasangan parameter masukan ke sistem (biasanya, sebuah algoritma perangkat lunak), tes semua kombinasi yang mungkin diskrit parameter tersebut. Menggunakan vektor uji dipilih dengan cermat, hal ini dapat dilakukan jauh lebih cepat daripada pencarian lengkap semua kombinasi dari semua parameter, dengan “parallelizing“ pengujian pasangan parameter. Jumlah tes biasanya  $O(nm)$ , dimana n dan m adalah jumlah kemungkinan untuk masing-masing dua parameter dengan pilihan yang paling.

Alasan di balik semua-All-pairs testing ini: yang sederhana dalam sebuah program umumnya dipicu oleh parameter masukan tunggal. Kategori paling sederhana berikutnya bug terdiri dari mereka bergantung pada interaksi antara pasangan parameter, yang bisa ditangkap dengan menguji semua-pasangan. yang melibatkan interaksi antara tiga atau lebih parameter secara progresif kurang umum [2], sementara pada saat yang sama waktu semakin lebih mahal untuk mencari oleh pengujian mendalam, yang sebagai batas pengujian lengkap semua input yang mungkin.

Banyak metode pengujian menganggap semua-pasang pengujian sistem atau subsistem sebagai kompromi biaya-manfaat yang wajar antara sering komputasi tidak layak tingkat tinggi metode pengujian kombinatorial, dan metode yang kurang lengkap yang gagal untuk menjalankan semua pasangan yang mungkin dari parameter. Karena tidak ada teknik pengujian dapat menemukan semua bug, semua-pasangan pengujian biasanya digunakan bersama dengan berbagai teknik jaminan

mutu seperti unit testing, eksekusi simbolik, pengujian bulu halus, dan memeriksa kode.

### *3. STATE TRANSITION TABLE*

Dalam teori automata dan logika sekuensial, state transition table adalah tabel yang menunjukkan apa yang negara (atau negara dalam kasus robot terbatas nondeterministic) suatu semiautomaton terbatas atau mesin finite state akan pindah ke, berdasarkan kondisi saat ini dan masukan lainnya. Sebuah tabel negara pada dasarnya adalah sebuah tabel kebenaran di mana beberapa input adalah kondisi saat ini, dan output termasuk negara berikutnya, bersama dengan keluaran lain.

state transition table adalah salah satu dari banyak cara untuk menentukan mesin negara, cara lain menjadi diagram negara, dan persamaan karakteristik.

### *4. EQUIVALENCE PARTITIONING*

Equivalence partitioning adalah pengujian perangkat lunak teknik yang membagi data masukan dari unit perangkat lunak menjadi beberapa partisi data dari mana test case dapat diturunkan. Pada prinsipnya, uji kasus dirancang untuk menutupi setiap partisi minimal sekali. Teknik ini mencoba untuk mendefinisikan kasus uji yang mengungkap kelas kesalahan, sehingga mengurangi jumlah kasus uji yang harus dikembangkan.

Dalam kasus yang jarang Equivalence partitioning juga diterapkan pada output dari komponen perangkat lunak, biasanya itu diterapkan pada masukan dari komponen diuji. Partisi ekivalen biasanya berasal dari spesifikasi persyaratan untuk atribut masukan yang mempengaruhi pengolahan benda uji. Sebuah masukan telah rentang tertentu yang rentang sah dan lainnya yang tidak valid. Data yang tidak valid di sini

tidak berarti bahwa data tidak benar, itu berarti bahwa data ini terletak diluar dari partisi tertentu. Hal ini mungkin lebih tepat dijelaskan oleh contoh fungsi yang mengambil sebuah parameter “bulan“. Jangkauan bulan adalah 1 sampai 12, mewakili Januari-Desember. Jangkauan ini disebut partisi. Dalam contoh ini ada dua partisi lebih lanjut rentang tidak valid. Partisi pertama akan menjadi tidak valid  $\leq 0$  dan partisi tidak valid kedua akan menjadi  $\geq 13$ .

### *5. BOUNDARY VALUES ANALYSIS*

Boundary value analysis merupakan suatu teknik pengujian perangkat lunak di mana tes dirancang untuk mencakup perwakilan dari nilai-nilai batas. Nilai-nilai di tepi sebuah partisi kesetaraan atau sebesar nilai terkecil di kedua sisi tepi. Nilai dapat berupa rentang masukan atau keluaran dari komponen perangkat lunak. Karena batas-batas tersebut adalah lokasi umum untuk kesalahan yang mengakibatkan kesalahan perangkat lunak mereka sering dilakukan dalam kasus-kasus uji.

Dokumentasi komponen software, mencangkup pemeriksaan dokumen dari software itu sendiri, yaitu :

- \* Flowchart yang dibuat
- \* Deskripsi input yang digunakan
- \* Deskripsi output yang digunakan
- \* Deskripsi output yang dihasilkan
- \* Kesesuaian penulisan (akurasi)
- \* Kontrol/kendali terhadap sistem yang dibuat

Strategi Black Box System, meliputi :

- \* **Batasan nilai untuk testing**, meliputi beberapa nilai, yaitu
  - Nilai minimum variabel input
  - Nilai di atas nilai minimum

- Nilai normal
  - Nilai di bawah nilai maksimum
  - Nilai maksimum
- \* **Equivalent Class Testing**, yaitu mengelompokkan input yang direpresentasikan sebagai hasil yang valid atau invalid. Contoh : Rekrutasi pegawai berdasarkan pengalaman kerja :
- <1thn : diterima, part time
- 1-3 thn : diterima, sebagai tenaga kerja profesional
- >4 thn : diterima, sebagai pegawai tetap

Kesalahan yang dapat terdeteksi melalui testing ini ialah :

- \* kebenaran dokumentasi
- \* akses basis data
- \* hasil akhir program

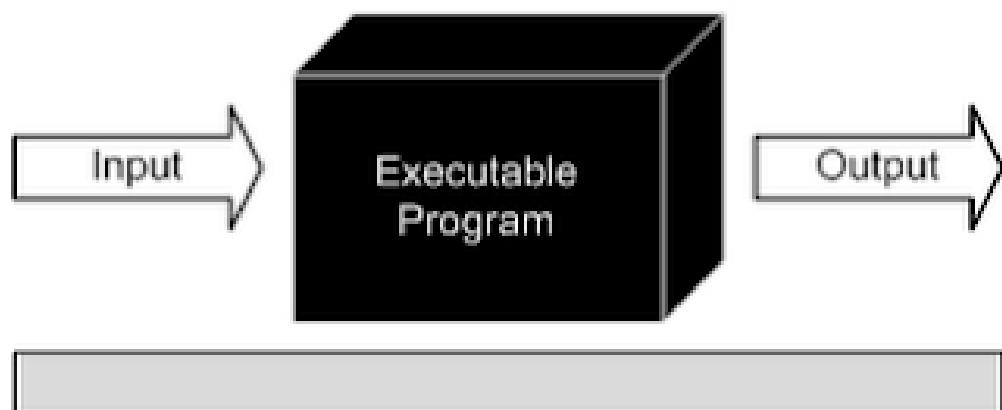
Kelebihan black box testing :

- \* Spesifikasi program dapat ditentukan di awal
- \* Dapat digunakan untuk menilai konsistensi program
- \* Testing dilakukan berdasarkan spesifikasi
- \* Tidak perlu melihat kode program secara detail

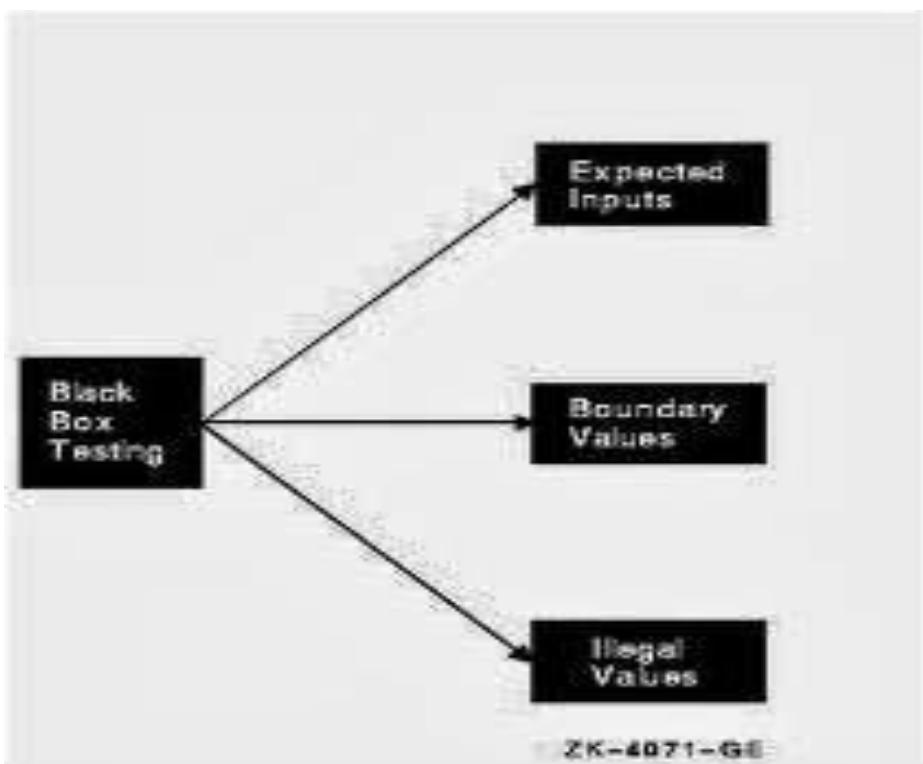
Kekurangan black box testing :

- \* Bila spesifikasi program yang dibuat kurang jelas dan ringkas, maka akan sulit membuat dokumentasi setepat mungkin

## BLACK BOX TESTING



*Gambar 4. 3 Alur Pengujian Blackbox*



Gambar 4. 4 Tahapan Pengujian Blackbox

### BLACKBOX TESTING

Metode ujicoba blackbox memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba

blackbox memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih

seluruh syarat-syarat fungsional suatu program. Ujicoba blackbox bukan merupakan alternatif dari ujicoba

whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain

menggunakan metode whitebox.

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

Pengujian adalah cara untuk mengetahui apakah sistem/aplikasi/alat yang dibuat sesuai dengan rancangan dan menuliskan hasil ujinya.

## 6.1 LINGKUNGAN IMPLEMENTASI

Perancangan aplikasi ini dapat berjalan dengan baik karena didukung oleh perangkat pendukung yaitu perangkat lunak dan perangkat keras.

### 6.1.1 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

Tabel 6. 1 Deskripsi Perangkat Lunak

No.	Jenis	Keterangan
1	Sistem Operasi	: Microsoft Windows 10
2	Bahasa Pemrograman	: PHP
3	<i>Database</i>	: MySQL
4	<i>Framework</i>	: CodeIgniter

### 6.1.2 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

Tabel 6. 2 Deskripsi Perangkat Keras

No.	Jenis	Keterangan
1	Prosesor	: Intel® Core™ i5
2	Memori (RAM)	: 4 GB
3	<i>Monitor</i>	: LCD 14,0 inci
4	<i>Mouse dan keyboard</i>	: Standard

## 1.2 6.2 PENGUJIAN DAN HASIL PENGUJIAN

### 1. 6.2.1 Identifikasi dan Rencana Pengujian

*Tabel 6. 3 Identifikasi Dan Rencana Pengujian*

Kelas Uji	Butir Uji	Identifikasi SKPL	Tingkat Pengujian	Jenis Pengujian	Jadwal
Validasi Login	Validasi Login Pelanggan	UC01	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
Tampil Form Utama	Tampil Menu Surat Masuk	UC02	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Tampil Menu Surat Keluar	UC03	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Tampil Menu Disposisi	UC04	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Tampil Menu Jadwal / Schedule	UC05	Pengujian Sistem	<i>Black Box</i>	10 – 12-2019
	Logout Bagian Umum dan Bidang	UC07	Pengujian Sistem	<i>Black Box</i>	25 – 09-2017

## 2. 6.2.2 Deskripsi dan Hasil Uji

### 6.2.2.1 Pengujian Validasi Login

*Tabel 6. 4 Pengujian Validasi Login*

<b>Identifikasi</b>	<b>Deskrripsi</b>	<b>Prosedur Pengujian</b>	<b>Masukan</b>	<b>Keluaran Yang Diharapkan</b>	<b>Kriteria Evaluasi Hasil</b>	<b>Hasil</b>	<b>Kesimpulan</b>
UC 01	Meng uji Login.	Klik Button OK	Masukkan User Name : Bagian Umum atau Passwor d :****	Tampil Form Utama Bagian Umum atau Bidang.	Tampil Form Utama Bagian Umum atau Bidang.	OK	Diterima

### 6.2.2.2 Pengujian Tampilan Utama Pelanggan

*Tabel 6. 5 Pengujian Tampilan Utama Pelanggan*

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil	Kesimpulan
UC02	Meng uji Menu Surat Masuk	Tampil Data	-Data Surat Masuk	- Data yang ada di database semuncul	Tampil di Halaman Surat Masuk	OK	Diterima
UC03	Meng uji Menu Surat Keluar	Tampil Data	-Data Surat Keluar	- Data yang ada di database semuncul	Tampil di Halaman Surat Keluar	OK	Diterima

<b>Identifikasi</b>	<b>Deskripsi</b>	<b>Prosedur Pengujian</b>	<b>Masukan</b>	<b>Keluaran yang Diharapkan</b>	<b>Kriteria Evaluasi Hasil</b>	<b>Hasil</b>	<b>Kesimpulan</b>
UC04	Meng uji Menu Delete Surat Masuk dan Keluar	Delete	-Data Surat	- Proses modifikasi berhasil kembali ke halaman Surat	Tampil halaman Surat	OK	Diterima
UC05	Meng uji Menu Jadwal / Schedule	Tampil Data	-Data Jadwal	- Data yang ada di database muncul	Tampil di Halaman Disposisi	OK	Diterima

### **6.2.2.3 Pengujian Logout**

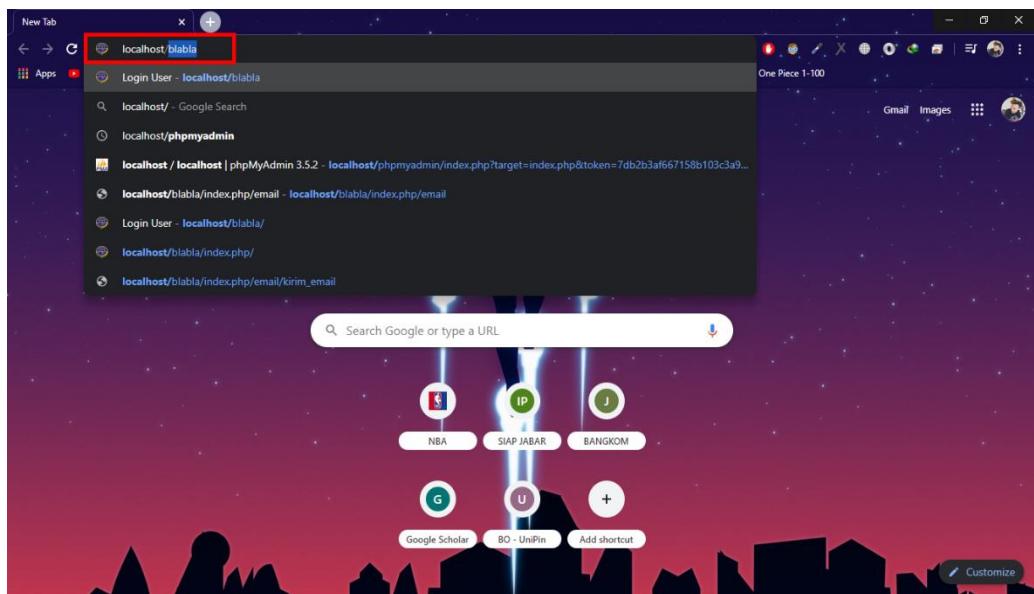
*Tabel 6. 6 Pengujian Logout*

<b>Identifikasi</b>	<b>Deskrripsi</b>	<b>Prosedur Pengujian</b>	<b>Masukan</b>	<b>Keluaran yang Diharapkan</b>	<b>Kriteria Evaluasi Hasil</b>	<b>Hasil</b>	<b>Kesimpulan</b>
SKPL11	Meng uji Logout	Klik menu logout lalu pilih logout	-	Tampil Form Bidang Umum atau Bidang	Tampil Form utama	OK	Diterima

## BAB VII

### 7.1 TUTORIAL PENGGUNAAN APLIKASI E-OFFICE

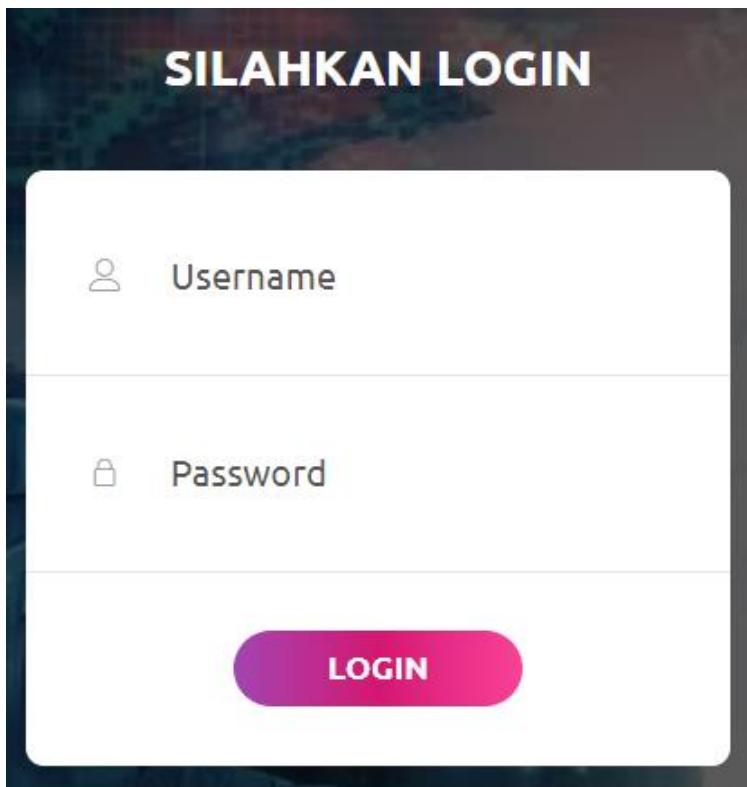
1. Buka browser kesayangan anda lalu ketikkan link aplikasi yang sudah anda buat, seperti ini :



*Gambar 7. 1 Penulisan URL*

Pada langkah ini anda sesuaikan link dengan nama folder aplikasi yang anda buat seperti gambar di atas disini penulis membuat nama folder menjadi “blabla” maka untuk membuka aplikasi tersebut pada browser ketikkan “localhost/blabla” atau “<http://localhost/blabla/>”, maka browser akan otomatis membuka aplikasi yang sudah anda buat, jangan lupa untuk menjalankan aplikasi xampp sebelum ada membuka aplikasi anda sendiri.

2. Menampilkan Halaman Login

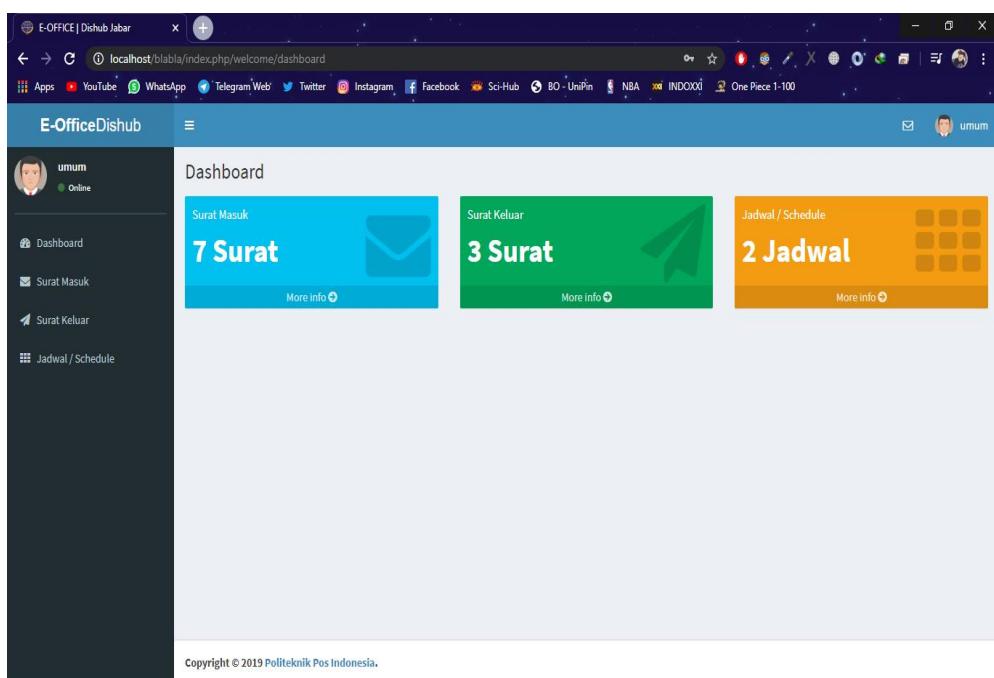


*Gambar 7. 2 Halaman Login*

Setelah menuliskan link atau url seperti langkah di atas dan dijalankan maka akan muncul halaman login aplikasi jika aplikasi anda menggunakan menu login namun jika tidak menggunakan menu login maka akan muncul halaman utama atau index aplikasi anda, disini penulis menggunakan menu login pada aplikasi yang di rancang oleh penulis dan jika aplikasi dijalankan maka akan muncul seperti gambar diatas, setelah menu login muncul maka silahkan masukkan username dan password pada form yang sudah disediakan, jika sudah

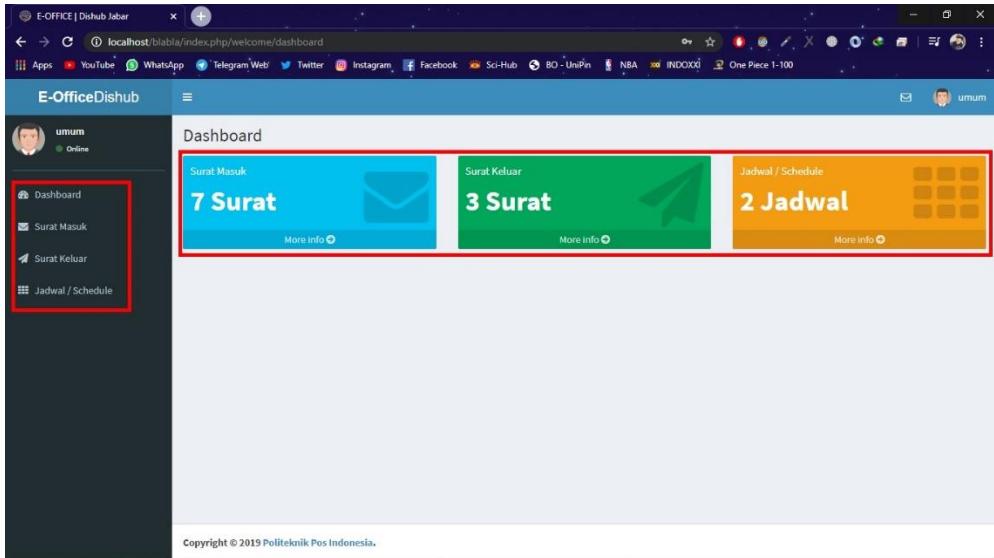
memasukkan username dan password maka system akan memeriksa username dan password tersebut pada database aplikasi jika username dan password yang dimasukkan sesuai dengan yang ada pada database maka aplikasi akan masuk ke menu selanjutnya, namun jika username dan password tidak sesuai dengan yang ada pada database maka system tidak akan melanjutkan ke halaman selanjutnya.

### 3. Halaman Dashboard



*Gambar 7. 3 Halaman Dasboard*

Jika sudah berhasil login dengan username dan password yang sesuai, maka akan masuk ke menu dashboard, tampilan menu dashboard seperti gambar diatas, pada menu dashboard terdapat tiga menu pada tampilan content dan 4 menu pada tampilan sidebar, jika ingin melanjutkan ke menu surat masuk, pembaca bias menekan menu yang terdapat pada tampilan konten atau tampilan menu pada sidebar.



Gambar 7. 4 Tampilan Dashboard dengan Kotak Merah

Pada gambar diatas anda bisa menekan di area kotak merah yang diberikan oleh penulis untuk melanjutkan ke menu selanjutnya.

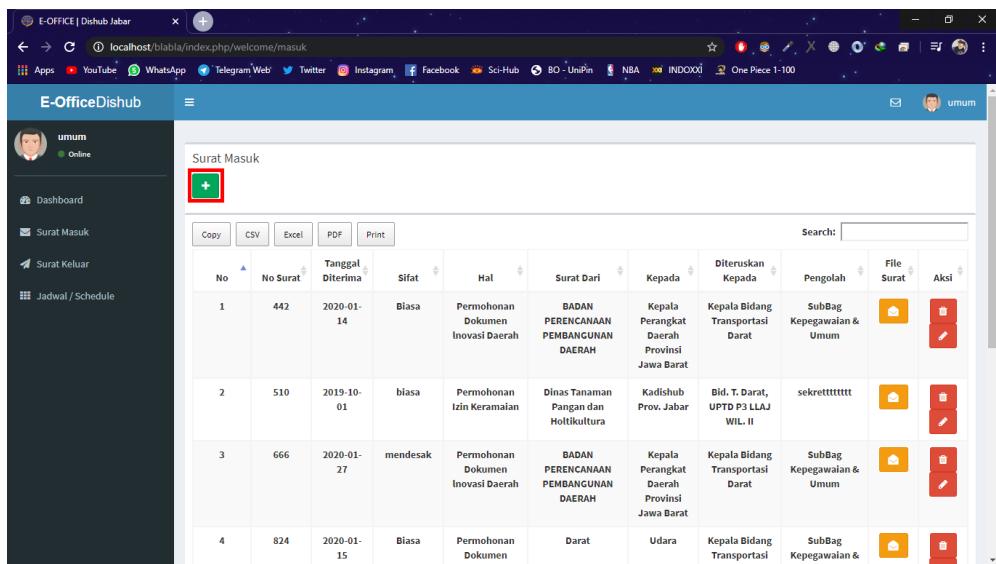
#### 4. Tampilan Surat Masuk

No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat	Aksi
1	442	2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawaian & Umum		
2	510	2019-10-01	biasa	Permohonan Izin Kramatan	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	Bid. T. Darat, UPTD P3 LLAJ WIL. II	sekretariat		
3	666	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawaian & Umum		
4	824	2020-01-15	Biasa	Permohonan Dokumen	Darat	Udara	Kepala Bidang Transportasi	SubBag Kepegawaian &		

Gambar 7. 5 Tampilan Menu Surat Masuk

Pada gambar diatas akan muncul ketika anda menekan menu “Surat masuk” tampilan ini akan menampilkan data surat masuk yang sudah dimasukkan pada aplikasi.

## 5. Menambah Surat Masuk



No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat	Aksi
1	442	2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepgawalian & Umum		
2	510	2019-10-01	biasa	Permohonan Izin Keramaian	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	Bid. T. Darat, UP TD P3 LLAJ WIL. II	sekretttttttt		
3	666	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepgawalian & Umum		
4	824	2020-01-15	Biasa	Permohonan Dokumen	Darat	Udara	Kepala Bidang Transportasi	SubBag Kepgawalian &		

Gambar 7. 6 Tampilan Surat Masuk dengan Kotak merah.

Pada halaman “Surat Masuk” jika anda ingin menambah surat masuk maka anda harus menekan tombol tambah surat masuk seperti pada gambar diatas yang sudah diberikan tanda merah oleh penulis jika anda menekan tombol itu maka akan melanjutkan ke halaman tambah surat masuk.

## 6. Menu Tambah Surat Masuk

The screenshot shows a web browser window titled 'E-OFFICE | Dishub Jabar'. The URL in the address bar is 'localhost:blabla/index.php/welcome/tambahmasuk/'. The page has a blue header with the title 'Tambah Surat Masuk'. On the left, there is a sidebar with user information ('umum', 'Online') and navigation links: 'Dashboard', 'Surat Masuk', 'Surat Keluar', and 'Jadwal / Schedule'. The main content area contains several input fields for adding a letter:

- Nomor Surat (Nomor Surat)
- Tanggal Diterima (mm/dd/yyyy)
- Sifat (Sifat)
- Isi Ringkas (Isi Ringkas)
- Surat Dari (Surat Dari)
- Kepada (Kepada)
- Diteruskan (Diteruskan)
- Pengolah (Pengolah)

*Gambar 7. 7 Tambah Surat Masuk*

Jika sudah menekan tombol tambah masuk maka akan menuju halaman tambah surat masuk seperti pada gambar diatas, setelah masuk pada halaman tambah surat masuk, maka inputkan semua data yang disiapkan pada form halaman surat masuk dalam proses penginputan tidak boleh ada satu field yang kosong karna jika kosong system tidak akan melanjutkan ke halaman utama surat masuk dan data surat tidak akan masuk pada system.

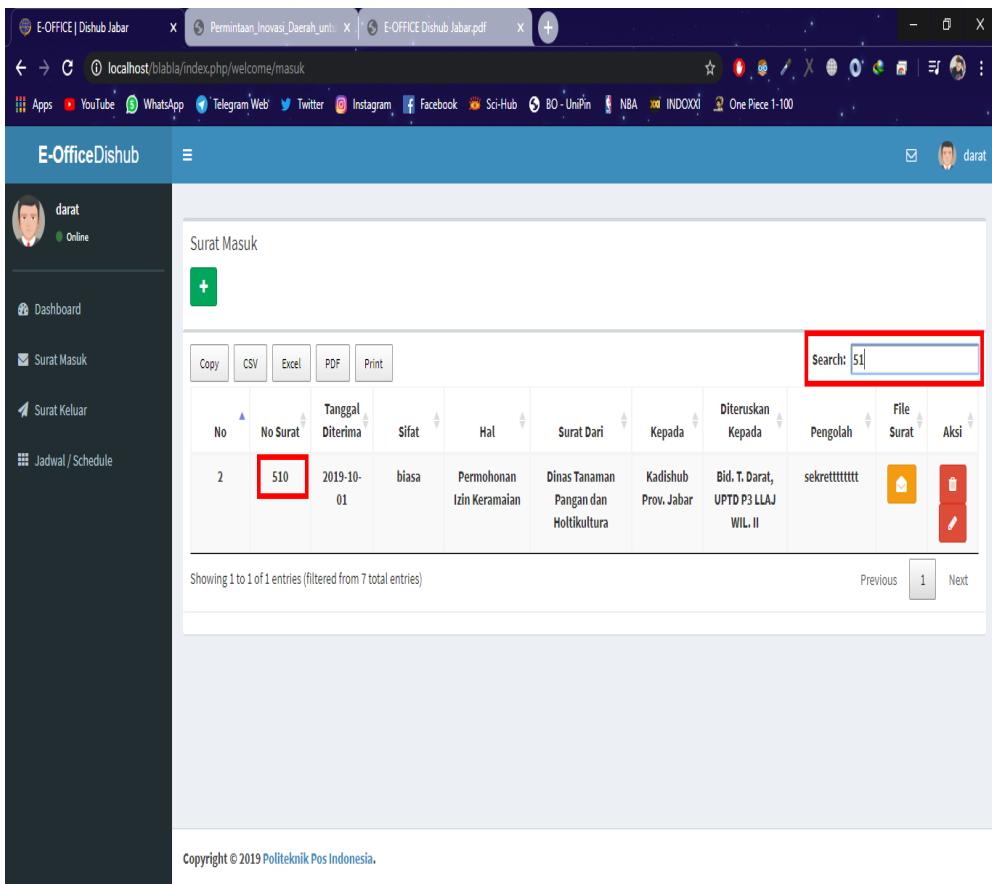
## 7. Mencari Surat masuk

The screenshot shows the E-Office Dishub application interface. On the left, there is a sidebar with user information ('umum', 'Online') and navigation links for 'Dashboard', 'Surat Masuk' (selected), 'Surat Keluar', and 'Jadwal / Schedule'. The main content area is titled 'Surat Masuk' and contains a table of incoming letters. At the top of the table are buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. To the right of the table is a 'Search:' input field with a magnifying glass icon, which is highlighted with a red box. The table has columns for 'No', 'No Surat', 'Tanggal Diterima', 'Sifat', 'Hal', 'Surat Dari', 'Kepada', 'Diteruskan Kepada', 'Pengolah', 'File Surat', and 'Aksi'. Each row in the table represents an incoming letter, with the first row being highlighted. The 'Aksi' column for each row contains three icons: a yellow folder (highlighted with a red box), a red trash can, and a red edit/pencil icon.

No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat	Aksi
1	442	2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawai dan Umum		
2	510	2019-10-01	biasa	Permohonan Izin Keramaian	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	Bid. T. Darat, UPTD P3 LLAJ WIL. II	sekretttttttt		
3	666	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawai dan Umum		
4	824	2020-01-15	Biasa	Permohonan Dokumen	Darat	Udara	Kepala Bidang Transportasi	SubBag Kepegawai dan Umum		

Gambar 7. 8 cari surat masuk

Pada halaman surat masuk, jika pembaca ingin mencari surat yang pembaca inginkan maka pembaca sudah disediakan menu pencarian dengan filter seperti pada gambar yang sudah diberikan tanda oleh penulis.



The screenshot shows a web-based application interface for managing incoming documents. On the left, there's a sidebar with user information ('darat', 'Online') and navigation links for 'Dashboard', 'Surat Masuk' (highlighted with a blue background), 'Surat Keluar', and 'Jadwal / Schedule'. The main content area is titled 'Surat Masuk' and displays a table of incoming letters. The table has columns: No, No Surat, Tanggal Diterima, Sifat, Hal, Surat Dari, Kepada, Diteruskan Kepada, Pengolah, File Surat, and Aksi. One row is visible, corresponding to the search result. At the top of the main area, there are buttons for Copy, CSV, Excel, PDF, Print, and a search input field containing '510' with a red border. Below the table, a message says 'Showing 1 to 1 of 1 entries (filtered from 7 total entries)'. At the bottom, there are navigation buttons for Previous, Next, and page number 1.

No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat	Aksi
2	510	2019-10-01	biasa	Permohonan Izin Keramaian	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	Bid. T. Darat, UPTD P3 LLAJ WIL. II	sekretttttttt		

Gambar 7. 9 Hasil Melakukan Pencarian Data Surat

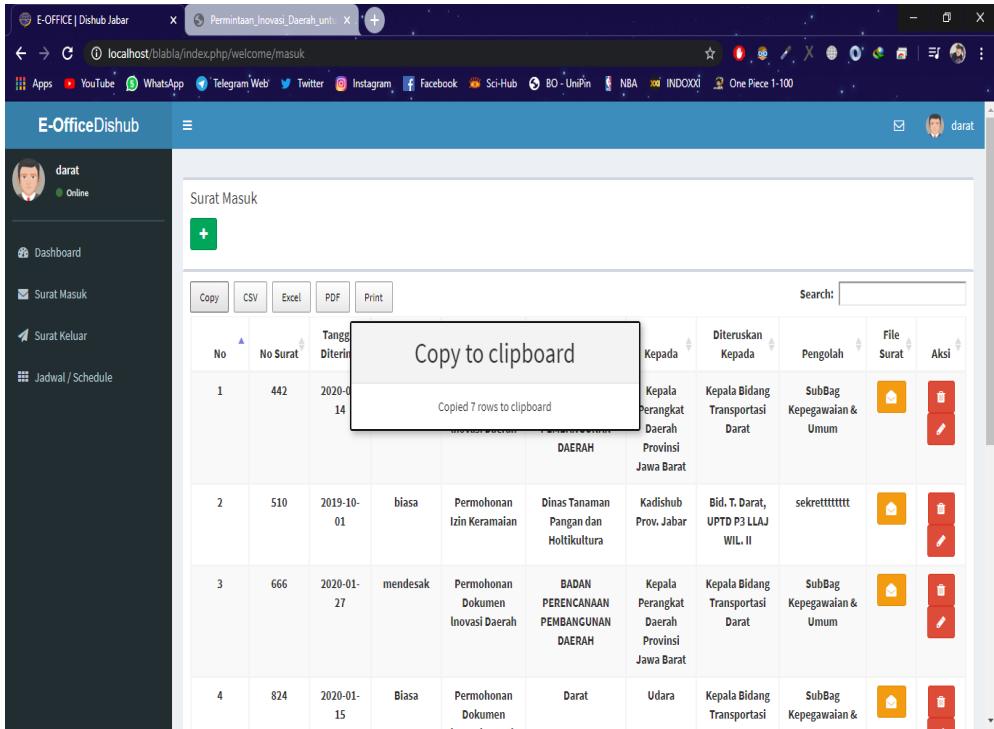
Pada gambar diatas adalah contoh ketika sudah melakukan pencarian suran, pada contoh diatas penulis melakukan pencarian dengan menggunakan no surat, maka akan muncul data surat dengan nomor yang sesuai dengan yang dituliskan pada menu pencarian.

## 8. Arsip Surat

No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat	Aksi
1	442	2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawaihan & Umum		
2	510	2019-10-01	biasa	Permohonan Izin Keramaian	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	Bid. T. Darat, UPTD P3 LLAJ WIL. II	sekretttttttt		
3	666	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawaihan & Umum		
4	824	2020-01-15	Biasa	Permohonan Dokumen	Darat	Udara	Kepala Bidang Transportasi	SubBag Kepegawaihan &		

Gambar 7. 10 Arsip Data Surat Dalam Beberapa bentuk File

Pada gambar diatas pembaca bisa mengunduh arsip dalam beberapa file, seperti yang sudah diberikan tanda pada gambar, file tersebut akan di ubah dalam bentuk csv, excel, pdf, dan bisa langsung di print tanpa harus mengunduh file arsipnya.



Gambar 7. 11 Tampilan Jika Menekan Tombol Copy

Pada gambar diatas memperlihatkan hasil dari aksi tombol copy yang berfungsi untuk menyalin data data arsip yang ada pada aplikasi tersebut, file tersebut akan di copy ke clipboard.

No	No Surat	Tanggal Diterir Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat
2	1	442	1/14/2020 Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN	E Kepala Pe Kepala Bidang Transport SubBag Kepegawaian & Umum		
3	2	510	10/1/2019 biasa	Permohonan Izin Keramaian	Dinas Tanaman Pangan dan Holtikultura Kadishub	Bid.T. Darat, UPTD P3 LI sekretttttttt		
4	3	666	1/27/2020 mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN	E Kepala Pe Kepala Bidang Transport SubBag Kepegawaian & Umum		
5	4	824	1/15/2020 Biasa	Permohonan Dokumen Inovasi Daerah	Darat	Udara Kepala Bidang Transport SubBag Kepegawaian & Umum		
6	5	1234	1/27/2020 mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN	E Kepala Pe Kepala Bidang Transport SubBag Kepegawaian & Umum		
7	6	445566	1/16/2020 penting	Permohonan Dokumen Inovasi Daerah	Kadishub	Kepala Pe Kepala Bidang Transport seke		
8	7	6664545	1/27/2020 mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN	E Kepala Pe Kepala Bidang Transport SubBag Kepegawaian & Umum		
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

Gambar 7. 12 Tampilan Arsip Dalam Bentuk CSV

Pada gambar diatas menampilkan hasil data surat yang diarsip dalam bentuk file csv. **Comma Separated Values** atau **CSV** adalah suatu format data dalam basis data di mana setiap *record* dipisahkan dengan tanda koma (,) atau titik koma (;). Selain sederhana, format ini dapat dibuka dengan berbagai *text-editor* seperti Notepad, Wordpad, ataupun MS.

	A	B	C	D	E	F	G
1	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	E-OFFICE   Dishub Jabar
2	1	442 2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	
3	2	510 2019-10-01	biasa	Permohonan Izin Keramahan	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	
4	3	666 2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	
5	4	824 2020-01-15	Biasa	Permohonan Dokumen Inovasi Daerah	Darat	Udara	
6	5	1234 2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	
7	6	445566 2020-01-16	penting	Permohonan Dokumen Inovasi Daerah	Kadishub	Kepala Perangkat Daerah Provinsi Jawa Barat	
8	7	6664545 2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							

Gambar 7. 13 Tampilan Arsip dalam Bentuk Excel

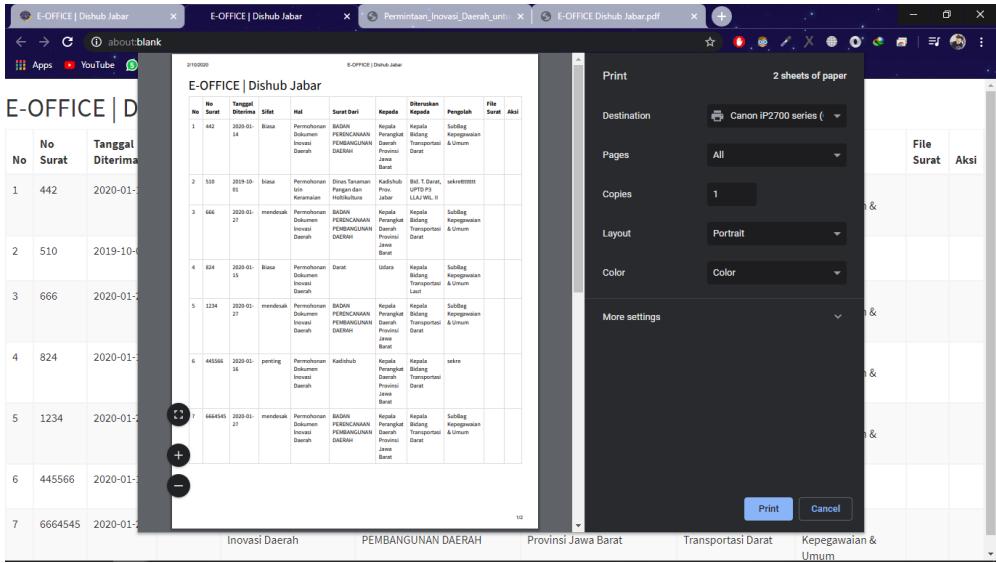
Pada gambar diatas menampilkan hasil data surat yang diarsip dalam bentuk file excel. **Microsoft Excel** atau **Microsoft Office Excel** adalah sebuah program aplikasi lembar kerja yang dibuat dan didistribusikan oleh Microsoft Corporation yang dapat dijalankan pada Microsoft Windows dan Mac OS. Aplikasi ini merupakan bagian dari Microsoft Office System. Aplikasi ini memiliki fitur kalkulasi dan pembuatan grafik yang, dengan menggunakan strategi *marketing* Microsoft yang agresif, menjadikan Microsoft Excel sebagai salah satu program komputer yang populer digunakan di dalam komputer mikro hingga saat ini. Bahkan, saat ini program ini merupakan program spreadsheet paling banyak digunakan oleh banyak pihak, baik di platform PC berbasis Windows maupun platform Macintosh berbasis Mac OS, semenjak versi 5.0 diterbitkan pada tahun 1993.

E-OFFICE   Dishub Jabar								
No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah
								A
1	442	2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi & Umum Darat	SubBag Kepegawaian
2	510	2019-10-01	biasa	Permohonan Izinan Keterlambatan	Dinas Tanaman Pangan dan Kehutanan	Kadishub Prov. Jawa Barat	Bid. T. Darat, sekretttttt	PTD P3 LLA Wil. II
3	666	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Kepegawaian	SubBag Transportasi & Umum Darat
4	824	2020-01-15	Biasa	Permohonan Dokumen Inovasi Daerah	Darat	Udara	Kepala Bidang Transportasi & Umum Laut	SubBag Kepegawaian
5	1234	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi & Umum Darat	SubBag Kepegawaian
6	445566	2020-01-16	penting	Permohonan Dokumen Inovasi	Kadishub	Kepala Perangkat Daerah	Kepala Bidang Transportasi	sekre

Gambar 7. 14 Tampilan Arsip dalam Bentuk PDF

Pada gambar diatas menampilkan hasil data surat yang diarsip dalam bentuk file pdf. PDF (*Portable Document Format*) adalah jenis format dokumen atau berkas untuk keperluan pertukaran dokumen digital yang dibuat oleh Adobe System pada tahun 1993.

Awalnya format ini digunakan untuk kepentingan pertukaran dokumen digital, namun sekarang format PDF juga banyak digunakan untuk presentasi dokumen dua dimensi yang terdiri dari huruf, teks, grafik vector dan citra.



*Gambar 7. 15 Tampilan Jika Menekan Tombol Print*

Pada gambar diatas menampilkan hasil data surat yang diarsip dalam bentuk file print. Disini anda dapat langsung mencetak file arsip tanpa harus mengunduh file tersebut.

## 9. Menampilkan File Surat Masuk

No	No Surat	Tanggal Diterima	Sifat	Hal	Surat Dari	Kepada	Diteruskan Kepada	Pengolah	File Surat	Aksi
1	442	2020-01-14	Biasa	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawai & Umum		
2	510	2019-10-01	biasa	Permohonan Izin Keramalan	Dinas Tanaman Pangan dan Holtikultura	Kadishub Prov. Jabar	Bid. T. Darat, UPTD P3 LLAJ WIL. II	sekretttttttt		
3	666	2020-01-27	mendesak	Permohonan Dokumen Inovasi Daerah	BADAN PERENCANAAN PEMBANGUNAN DAERAH	Kepala Perangkat Daerah Provinsi Jawa Barat	Kepala Bidang Transportasi Darat	SubBag Kepegawai & Umum		
4	824	2020-01-15	Biasa	Permohonan Dokumen	Darat	Udara	Kepala Bidang Transportasi	SubBag Kepegawai &		

Gambar 7. 16 Button Menampilkan File Surat Masuk

Pada gambar diatas anda dapat melihat isi file surat yang sudah dimasukkan pada aplikasi tersebut dengan cara menekan tombol dengan gambar amplop terbuka seperti yang sudah ditandai dengan warna merah oleh penulis.



*Gambar 7. 17 Tampilan File Surat*

Pada gambar diatas menampilkan isi file surat setelah menekan tombol yang dianjurkan seperti langkah sebelumnya.

## 10. Tampilan Jadwal

No	Tanggal	Kegiatan	Waktu	Tempat	Dihadiri Oleh	Aksi
1	2020-01-17	Rapat Pimpinan	10.30 - Selesai	Gedung Sate	Kadishub Dinas Perhubungan Prov. Jabar	
2	0000-00-00	Rapat Pimpinan	09.00 - 10.30	Ruang Rapat N250	Kepala Bagian & Kepala Bidang	

*Gambar 7. 18 Tampilan Jadwal*

## 11. Tampilan Tambah Jadwal

The screenshot shows a web-based application titled 'E-OfficeDishub'. On the left, there is a sidebar with a user profile picture and the name 'umum' with an 'Online' status. Below the profile are four menu items: 'Dashboard', 'Surat Masuk', 'Surat Keluar', and 'Jadwal / Schedule'. The main content area is titled 'Tambah Jadwal' and contains five input fields: 'Tanggal' (Date), 'Kegiatan' (Activity), 'Waktu' (Time), 'Tempat' (Place), and 'Disposisi' (Disposition). Each field has a placeholder text and a corresponding input box. A blue 'Tambah' (Add) button is located at the bottom left of the form. At the very bottom of the page, there is a copyright notice: 'Copyright © 2019 Politeknik Pos Indonesia.'

*Gambar 7. 19 Tampilan Tambah Jadwal*