

BAB 4

PEMBAHASAN DAN HASIL PENELITIAN

4.1 Perancangan Aplikasi

Aplikasi *Virtual Assistant* Berbasis Hologram dalam Statistik Pengunjung di Perpustakaan Unika Soegijapranata dibuat dengan tujuan melakukan pencatatan statistik pengunjung perpustakaan, dan pencarian buku berdasarkan judul, pengarang dan topik. Dikemas dengan *User Interface* yang sederhana namun futuristik, agar mahasiswa mudah memahami dan menarik untuk dimanfaatkan.

Aplikasi *Virtual Assistant* dirancang melalui beberapa tahap, yaitu:

1. Perancangan perangkat keras, meliputi:
 - a. perancangan layar hologram
 - b. Perancangan *booth*, dan
 - c. Penggabungan setiap perangkat keras menjadi satu.
2. Perancangan perangkat lunak, meliputi:
 - a. Pembuatan tampilan layar statistik
 - b. Pembuatan tampilan pencarian buku, dan
 - c. Pembuatan tampilan diam.

Aplikasi *Virtual Assistant* ini berbasis web lokal sehingga aplikasi harus terinstall dalam komputer (CPU). Dalam aplikasi *Virtual Assistant* ini terdapat fitur *text-to-speech* yakni mengubah teks menjadi suara untuk menambahkan kesan “*Virtual Assistant*” didalamnya. Berikut gambar di bawah ini adalah *Use Case Diagram* perancangan aplikasi *Virtual Assistant* HoLi (*Hologram Library*)

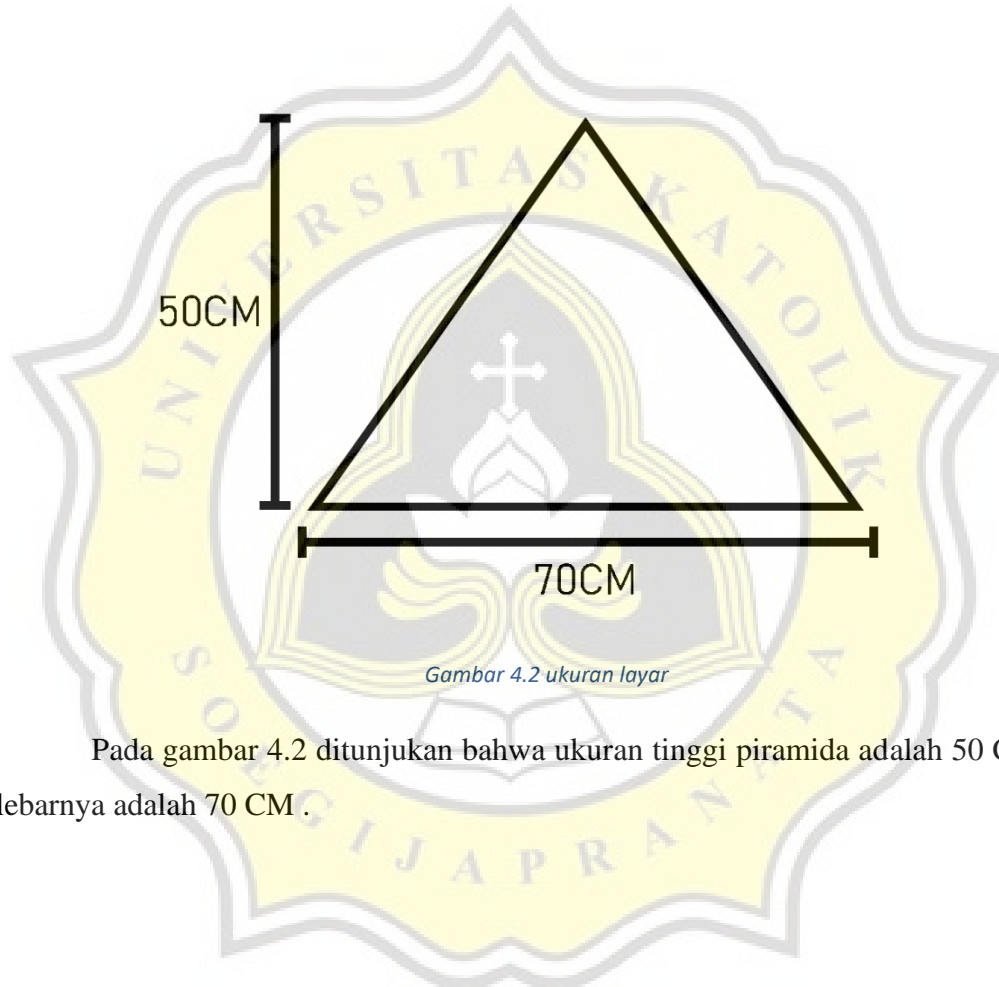


Perancangan dimulai dengan menentukan bahasa pemrograman yang akan dipakai. Dengan melalui berbagai diskusi disimpulkan bahasa yang digunakan adalah HTML, PHP, CSS, dan Javascript karena platform yang paling cocok untuk mengelola data dan fitur *text-to-speech* adalah pemrograman web.

Perancangan aplikasi dibagi menjadi 2 bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak:

a. Perancangan Perangkat Keras

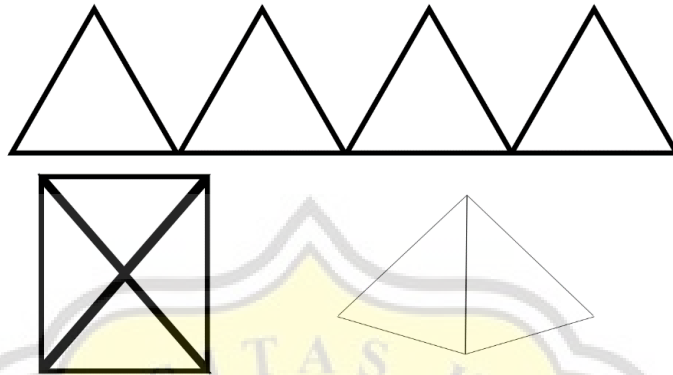
Piramida dirancang sesuai dengan ukuran layar yang akan digunakan, kemudian ditetapkan sesuai dengan dimensi layar seperti gambar di bawah ini.



Gambar 4.2 ukuran layar

Pada gambar 4.2 ditunjukkan bahwa ukuran tinggi piramida adalah 50 CM dan lebarnya adalah 70 CM .

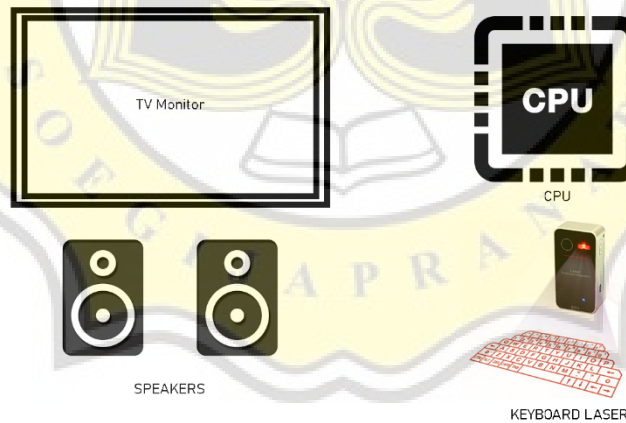
Pada gambar di bawah adalah pemetaan dari bentuk piramida yang akan digunakan sebagai layar dari hologram.



Gambar 4.3 piramida

Layar kemudian digabungkan sehingga membentuk piramida seperti pada gambar 4.3 sehingga layar terbagi menjadi 4 bagian.

Pada gambar di 4.4 di bawah ini adalah perangkat keras yang digunakan oleh sistem Virtual Assistant.

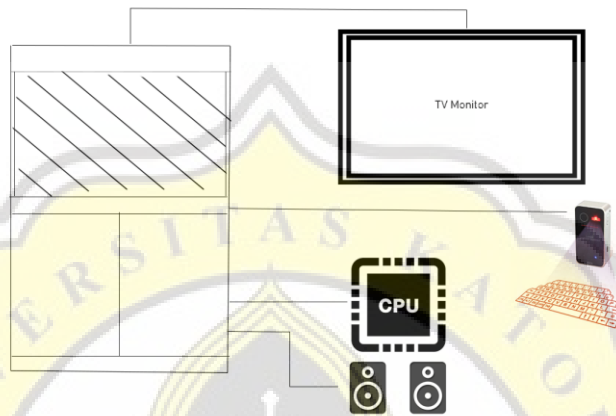


Gambar 4.4 peralatan

Macam-macam peralatan yang dibutuhkan yaitu TV Monitor untuk menampilkan aplikasi, CPU untuk menjalankan aplikasi, speaker untuk memutar suara

text-to-speech, dan keyboard laser untuk mengetik input dan pelengkap agar terkesan modern.

Gambar di bawah adalah pemetaan dari penempatan perangkat keras pada sistem.



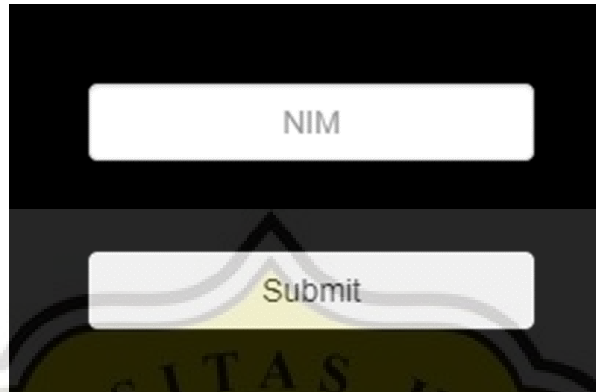
Gambar 4.5 penggabungan

Kemudian semua perangkat yang tersedia disatukan dalam satu *booth* untuk CPU dan Speaker diletakan di dalam *booth* agar tetap mempertahankan estetika dari aplikasi. Keyboard laser diletakan di tengah agar pengguna mudah untuk meraihnya, dan TV monitor diletakan di atas dengan layar menghadap ke bawah, sehingga pantulan cahaya dari TV monitor diterima oleh piramida hologram.

b. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini pertama dengan membangun tampilan *input* bagi pengguna untuk memasukan NIM pada aplikasi *Virtual Assistant*. Pada gambar di bawah adalah tampilan utama yang disebut sebagai layar *Input NIM*, di mana pengguna akan memasukan NIM mereka untuk masuk sebagai pengunjung di perpustakaan.

Gambar 4.6 adalah halaman utama yang pengguna akan lihat pada layar hologram untuk mengisi daftar kehadiran di perpustakaan.



Gambar 4.6 Input NIM

Aplikasi *Virtual Assistant* dimulai dengan layar input NIM di mana pengguna akan memasukan NIM kemudian tekan Enter untuk submit.

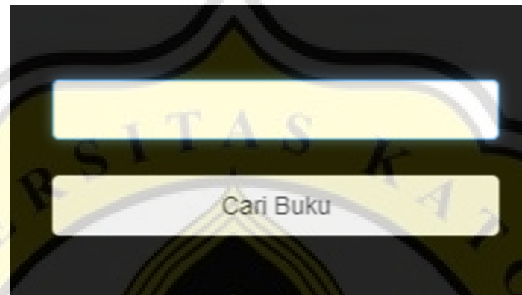
Gambar di bawah ini adalah hasil dari *input* pengguna, dengan dikeluarkan teks menyambut nama dari pengguna dan menampilkan wajah dari karakter *Visual Assistant*.



Gambar 4.7 Output NIM

Setelah NIM dimasukan, maka akan mengalihkan laman ke karakter *Virtual Assistant* yang kemudian dilanjutkan dengan berbicara “Selamat Datang + nama mahasiswa + Selamat Berliterasi”. Setelah karakter menyapa pengguna kemudian laman akan kembali ke laman Input Nim.

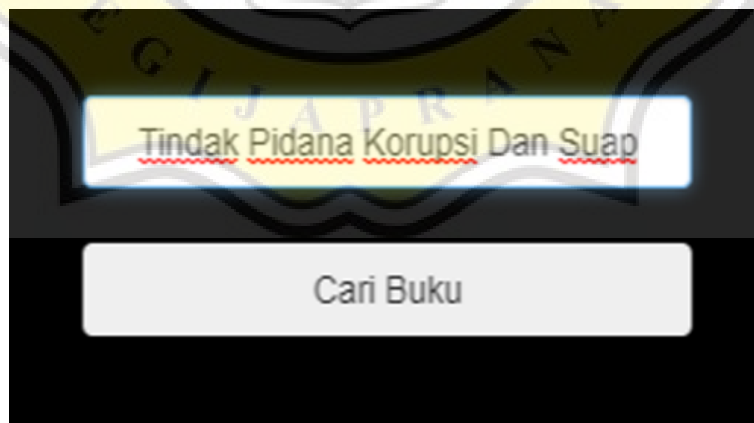
Pada gambar 4.8 menunjukkan tampilan lain yang berfungsi untuk mencari lokasi buku di sekitar perpustakaan Unika.



Gambar 4.8 Cari Buku

Jika menekan tombol Insert pada keyboard di laman statistik pengunjung maka akan memindahkan laman statistik pengunjung ke laman “Cari Buku”. Pada laman ini pengguna dapat mencari buku berdasarkan judul, nama pengarang dan topik.

Pada gambar 4.9 adalah tampilan jika pengguna melakukan pencarian pada halaman “Cari Buku”



Gambar 4.9 Input Berdasarkan Judul

Jika input yang dimasukan berdasarkan judul seperti gambar 4.9 maka output akan keluar seperti gambar 4.10

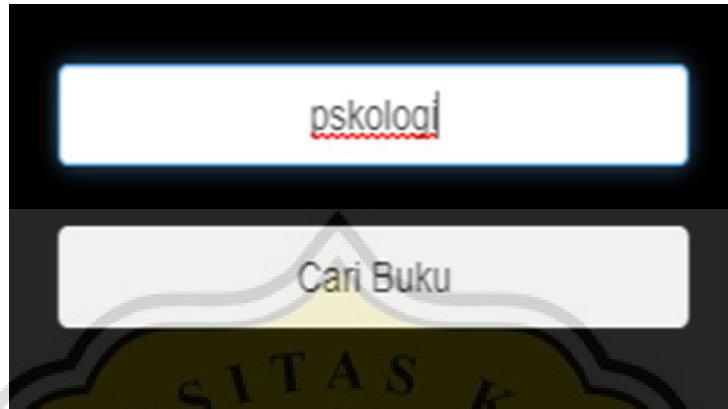
Gambar di bawah ini adalah contoh dari hasil pencarian yang dilakukan oleh pengguna.



Gambar 4.10 Output Judul

Output hanya akan memunculkan satu hasil yang memuat nama buku, dan lokasi buku serta Virtual Assistant akan membacakannya kepada pengguna.

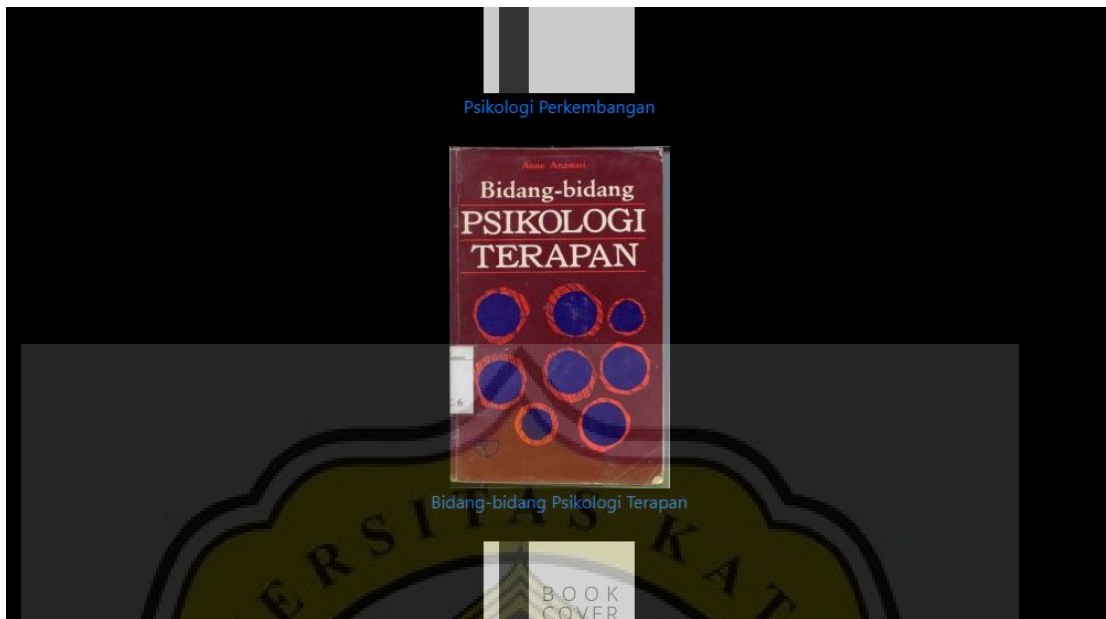
Gambar 4.11 adalah contoh dari pencarian buku dengan cara yang lain yaitu berdasarkan topik.



Gambar 4.11 Input Topik atau pengarang

Tidak semua pengguna tahu judul sebuah buku yang hendak mereka cari, sehingga aplikasi Virtual Assistant menyediakan pencarian berdasarkan nama pengarang atau topik. Jika pengguna memasukkan nama topik atau pengarang maka hasil yang dikeluarkan akan seperti pada gambar 4.12.

Gambar 4.12 adalah tampilan jika pencarian yang dilakukan oleh pengguna memiliki hasil lebih dari satu.



Gambar 4.12 Output Topik atau pengarang

Pada gambar 4.12 adalah daftar dari buku yang memiliki topik “psikologi” sehingga pengguna dapat memilih buku yang sedang mereka cari .

Gambar di bawah ini adalah tampilan hasil dari pencarian buku tersebut.



Gambar 4.13 Output Pencarian

Setelah pengguna memilih buku yang dicari maka laman akan berpindah ke karakter Virtual Assistant yang menunjukkan lokasi buku tersebut.

Gambar 4.14 adalah tampilan dari halaman *idle*.



Gambar 4.14 Laman Idle

Jika tidak ada aktivitas pada aplikasi Virtual Assistant dalam waktu 30 detik maka laman akan pindah ke laman *idle* yang digunakan untuk mempromosikan Virtual library.

4.2 Pembuatan Aplikasi

a. Layar Statistik Pengunjung

Gambar di bawah adalah tampilan dari halaman statistik pengunjung

A screenshot of a form on a black background. It features two white rectangular input fields. The top field is labeled "NIM" and the bottom field is labeled "Submit".

Gambar 4.15 Input NIM

Dimulai dengan membuat sebuah *form* sederhana seperti gambar 4.15 di mana terdapat *form input* dan tombol *submit* di bawahnya.

Gambar di bawah ini adalah kode dari halaman statistik pengunjung dengan format HTML.

```
<form id="form3" class="form" onsubmit="Simpan()">
  <input type="text" name="id_anggota3" id="id_anggota3" class="center
  form-control" placeholder="NIM" autocomplete="off">
  <br><br>
  <input type="hidden" class="form-control" id="institusi" name="
  institusi" placeholder="Institusi" autocomplete="off">
  <input type="submit" name="submit" value="Submit" class="input-button
  btn btn-secondary">
</form>
```

gambar 4.16 Kode dari Halaman Statistik Pengunjung

Pada gambar 4.16 terdapat kode dari halaman statistik pengunjung berdasarkan *id_anggota* serta *function input* yang bertipe *submit*.

Pada gambar 4.17 adalah kode dari *form input* yang ditetapkan agar memunculkan keempat sisi *form* pada layar hologram. Untuk *div-1* akan memunculkan *form* statistik pengunjung pada sisi belakang, dan *div-2* pada sisi kanan di layar hologram.

```
<body class="bg-color">
  <div class="full-body">
    <div class="div-content">
      <!-- Atas -->
      <div class="div-1">
        <form id="form3" class="form" onsubmit="Simpan()">
          <input type="text" name="id_anggota3" id="id_anggota3" class="center form-control" placeholder="NIM"
          autocomplete="off">
          <br><br>
          <input type="hidden" class="form-control" id="institusi" name="institusi" placeholder="Institusi"
          autocomplete="off">
          <input type="submit" name="submit" value="Submit" class="input-button btn btn-secondary">
        </form>
        <video id="video3" width="320" height="240" style="display: none;">
          <source src="selamatdatangv2.mp4" type="video/mp4">
        </video>
      </div>
      <!-- Kanan -->
      <div class="div-2">
        <form id="form2" class="form" onsubmit="Simpan()">
          <input type="text" name="id_anggota4" id="id_anggota4" class="center form-control" placeholder="NIM"
          autocomplete="off">
          <br><br>
          <input type="hidden" class="form-control" id="institusi2" name="institusi2" placeholder="Institusi"
          autocomplete="off">
          <input type="submit" name="submit" value="Submit" class="input-button btn btn-secondary">
        </form>
        <video id="video2" width="320" height="240" style="display: none;">
          <source src="selamatdatangv2.mp4" type="video/mp4">
        </video>
      </div>
    </div>
  </div>
</body>
```

Gambar 4.17 Form Input Atas dan Kanan

Gambar di bawah ini adalah kode yang sama seperti pada gambar 4.17 hanya saja pembagian layarnya yang berbeda. Pada div-3 layar statistik pengunjung akan ditampilkan pada sisi depan, dan div-4 akan ditampilkan pada sisi kiri.

```
<!-- Bawah -->
<div class="div-3">
  <form method="POST" id="form1" class="form" onsubmit="return Simpan()" style="display: block;">
    <input type="text" name="id_anggota1" id="id_anggota1" autofocus class="center form-control" placeholder="NIM" autocomplete="off" onkeyup="ketikSemua()" onchange="Simpan()">
    <br><br>
    <input type="hidden" class="form-control" id="institusi3" name="institusi3" placeholder="Institusi" autocomplete="off">
    <input type="submit" name="submit" value="Submit" class="input-button btn btn-secondary">
  </form>
  <video id="video1" width="320" height="240" style="display: none;">
    <source src="selamatdatangv2.mp4" type="video/mp4">
  </video>
</div>
<!-- Kiri -->
<div class="div-4">
  <form id="form4" class="form" onsubmit="Simpan()">
    <input type="text" name="id_anggota2" id="id_anggota2" class="center form-control" placeholder="NIM" autocomplete="off">
    <br><br>
    <input type="hidden" class="form-control" id="institusi4" name="institusi4" placeholder="Institusi" autocomplete="off">
    <input type="submit" name="submit" value="Submit" class="input-button btn btn-secondary">
  </form>
  <video id="video4" width="320" height="240" style="display: none;">
    <source src="selamatdatangv2.mp4" type="video/mp4">
  </video>
</div>
```

Gambar 4.18 Form Input Bawah dan Kiri

Seperti pada gambar 4.18 di dalam *class div-3* terdapat metode *ketikSemua()* agar setiap *form* yang terdapat pada keempat sisi akan terisi secara bersamaan berdasarkan *input* dari *class div-3*.

Gambar di bawah ini adalah kode CSS pada halaman statistik pengunjung.

```
.bg-color{
  background-color: black;
}
.center{
  text-align: center;
}
.full-body{
  height: 100%;
  width: 100%;
  display: grid;
  grid-template-columns: 22.2% 55.5% 22.2%;
  grid-template-rows: 100%;
}
.div-content{
  grid-column-start: 2;
  grid-column-end: 3;
  display: grid;
  grid-template-columns: 33.3% 33.3% 33.3%;
  grid-template-rows: 33.3% 33.3% 33.3%;
}
```

Gambar 4.19 kode pemrograman CSS Layar Virtual Assistant

Gambar 4.19 menunjukkan kode pemrograman untuk mengelola *background* secara visual. *.bg-color* berfungsi untuk mengganti warna *background* menjadi warna hitam untuk semua laman aplikasi *Virtual Assistant*. Diteruskan dengan *.center* yang berfungsi agar semua teks menjadi di tengah. Kemudian *.full-body* berfungsi untuk membagi layar menjadi 3 bagian yaitu bagian kiri dan kanan 22.2% dan untuk bagian tengah 55.5% seperti pada gambar 4 20. Setelah itu *.div-content* untuk membagi setiap layar agar tiap *form input* memiliki bagiannya sendiri.

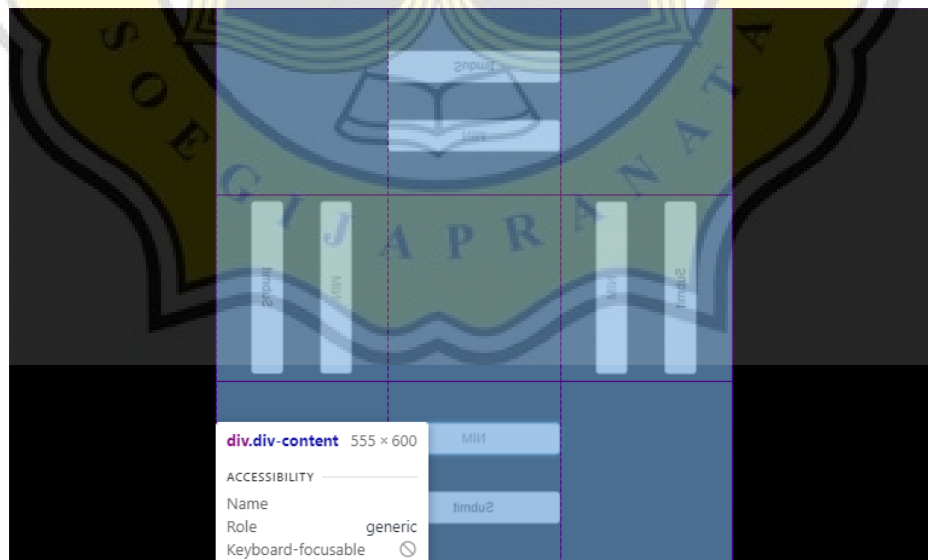
Gambar 4.20 adalah pembagian dimensi dari halaman statistik pengunjung



Gambar 4.20 Pembagian Layar Melalui .full-body

Pada gambar 4.20 fungsi dari *full-body* membagi layar menjadi 3 agar konten aplikasi *Virtual Assistant* memiliki area berbentuk persegi, agar *output* yang keluar dari layar TV Monitor akan sesuai dengan layar hologram yang berbentuk piramida.

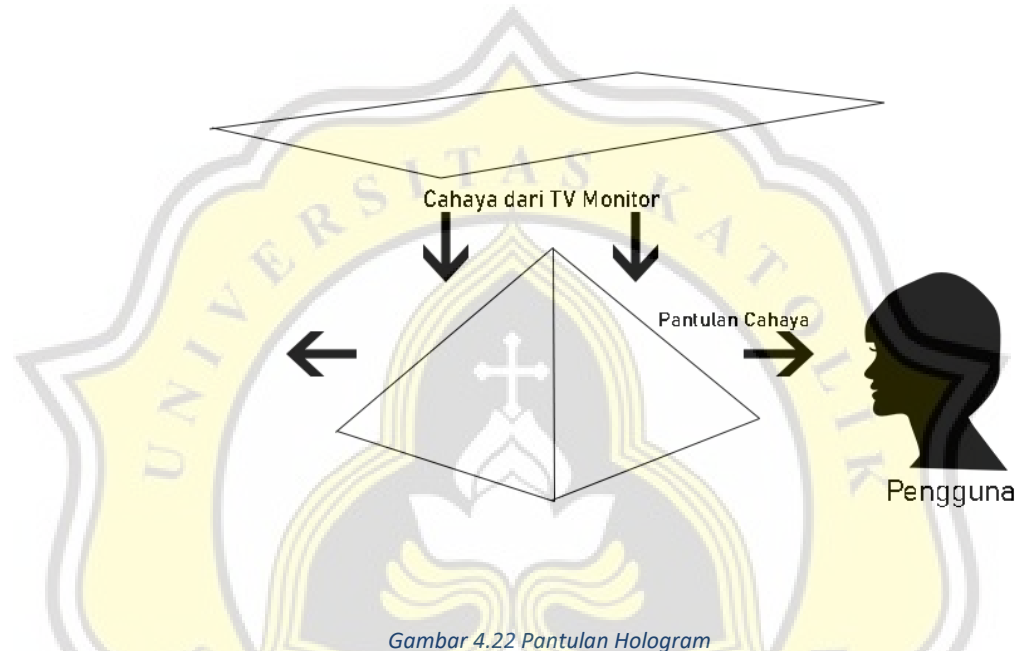
Gambar di bawah ini adalah pembagian dari keempat sisi halaman statistik pengunjung yang telah disesuaikan dengan layar hologram



Gambar 4.21 Pembagian Layar Melalui div-content

Gambar 4.21 menunjukkan pembagian layar melalui fungsi *div-content*. Layar dibagi menjadi 9 bagian agar *form input* dapat menyesuaikan layar TV Monitor.

Pada gambar di bawah ini adalah model dari pantulan hologram dari layar monitor hingga diterima oleh pengguna.



Gambar 4.22 menunjukkan bahwa visualisasi yang diterima oleh pengguna adalah cahaya yang dipantulkan oleh layar hologram. Sehingga yang diterima oleh pengguna adalah pantulan seperti cermin.

Pada gambar 4.23 ini adalah kode CSS untuk memutar dan menempatkan *form* statistik pengunjung pada *form input class div-1*.


```

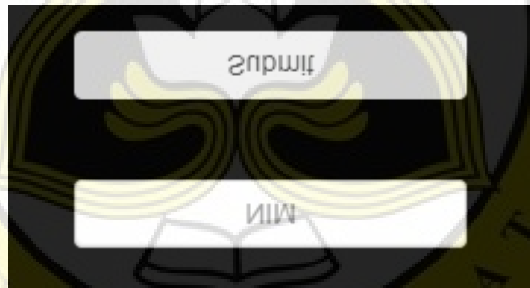

.div-1{
  grid-column-start: 2;
  grid-column-end: 3;
  grid-row-start: 1;
  grid-row-end: 2;
  margin: auto;
  transform: rotateY(180deg) rotateZ(180deg);
}


```

gambar 4.23 Membagi Lokasi Form div-1

Gambar 4.23 menunjukkan bahwa *form* yang memiliki *class* bernama *div-1* akan menempati kolom ketiga dan baris kedua yang artinya *form div-1* akan ditempatkan pada tengah atas. Dilanjutkan oleh *transform* yang dibagi menjadi 2 yaitu *rotateY* dan *rotateZ*. *rotateZ* memutar axis Z pada form sebesar 180 derajat, sedangkan *rotateY* memutar axis Y sebesar 180 derajat sehingga memberikan efek cermin pada aplikasi *Virtual Assistant*.

Gambar di bawah ini adalah tampilan untuk sisi depan pada layar hologram.



Gambar 4.24 Form NIM

Pada gambar 4.24 adalah hasil dari efek cermin yang dikeluarkan oleh layar TV Monitor sehingga ketika dipantulkan melalui layar hologram, *form* menjadi terbaca oleh pengguna.

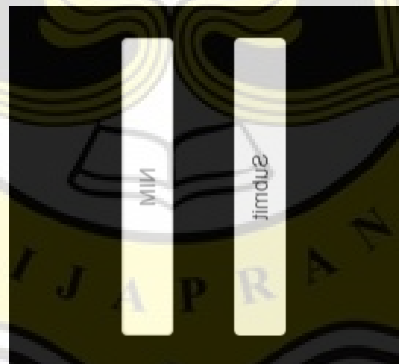
Pada gambar 4.25 ini adalah kode CSS untuk memutar dan menempatkan *form* statistik pengunjung berdasarkan kolom dan baris pada *form input class div-2*.

```
.div-2{  
  grid-column-start: 3;  
  grid-column-end: 4;  
  grid-row-start: 2;  
  grid-row-end: 3;  
  margin: auto;  
  transform: rotateY(180deg) rotateZ(90deg);  
}
```

gambar 4.25 Membagi Lokasi Form div-2

Selanjutnya pada *div-2* memposisikan form pada tengah-kanan, kemudian memutar axis Z sebesar 90 derajat. Sehingga membuat *form* dengan *class div-2* akan miring 90 derajat ke arah kanan, seperti pada gambar 4.26.

Gambar di bawah ini adalah tampilan untuk sisi kiri pada layar hologram.



Gambar 4.26 Form Class div-2

Gambar 4.26 menunjukkan *form* dengan *class div-2*, fungsinya agar pengunjung dapat melihat aplikasi *Virtual Assistant* dari sisi kiri layar hologram.

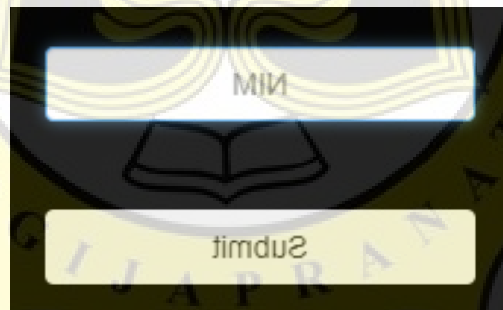
Pada gambar 4.27 ini adalah kode CSS untuk memutar dan menempatkan *form* statistik pengunjung berdasarkan kolom dan baris pada *form input class div-3*.

```
.div-3{
  grid-column-start: 2;
  grid-column-end: 3;
  grid-row-start: 3;
  grid-row-end: 4;
  margin: auto;
  transform: rotateY(180deg);
  text-decoration-color: white;
}
```

Gambar 4.27 Form Class div-3

Pada gambar 4.27 menunjukkan bahwa *form* dengan *class* div-3 akan menempatkan posisinya pada tengah-bawah, dan seperti *form* yang lain, *class* div-3 akan memberikan efek cermin pada *form* seperti pada gambar di bawah ini.

Gambar di bawah ini adalah tampilan untuk sisi belakang pada layar hologram.



Gambar 4.28 Form dengan Class div-3

Gambar 4.28 menunjukkan *form* dengan *class* div-3, agar pengunjung dapat melihat aplikasi Virtual Assistant dari belakang.

Pada gambar 4.29 ini adalah kode CSS untuk memutar dan menempatkan *form* statistik pengunjung berdasarkan kolom dan baris pada *form input class div-4*.

```
.div-4{
  grid-column-start: 1;
  grid-column-end: 2;
  grid-row-start: 2;
  grid-row-end: 3;
  margin: auto;
  transform: rotateY(180deg) rotateZ(-90deg);
}
```

Gambar 4.29 Form Class div-4

Selanjutnya yang terakhir membuat *style div-4*, *form* dengan *class div-4* akan ditempatkan pada sisi kiri layar TV Monitor dan memutar axis Z sejumlah -90 derajat. Sehingga menghasilkan *form* menjadi seperti gambar 4.30.

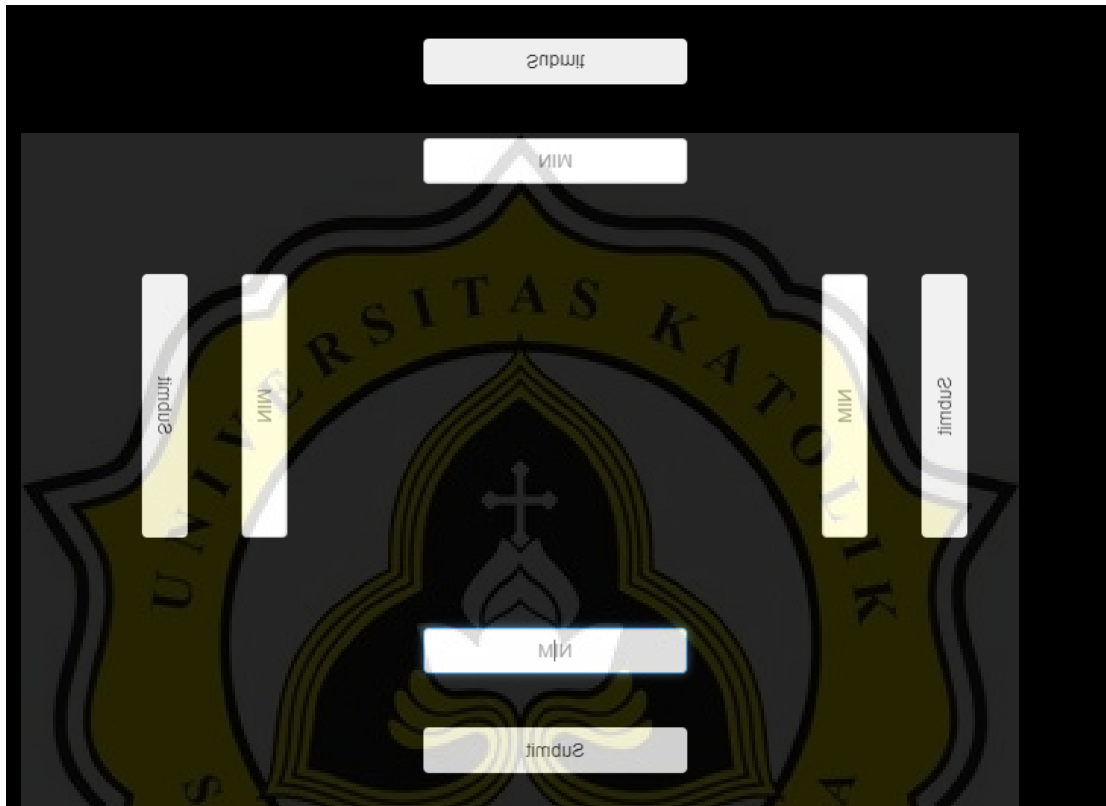
Gambar di bawah ini adalah tampilan untuk sisi kanan pada layar hologram.



Gambar 4.30 Form dengan class div-4

Fungsi dari *form* dengan *class div-4* adalah agar pengguna dari arah sisi kanan dapat melihat hologram dari aplikasi Virtual Assistant tersebut.

Gambar di bawah ini adalah tampilan secara keseluruhan dari halaman statistik pengunjung.

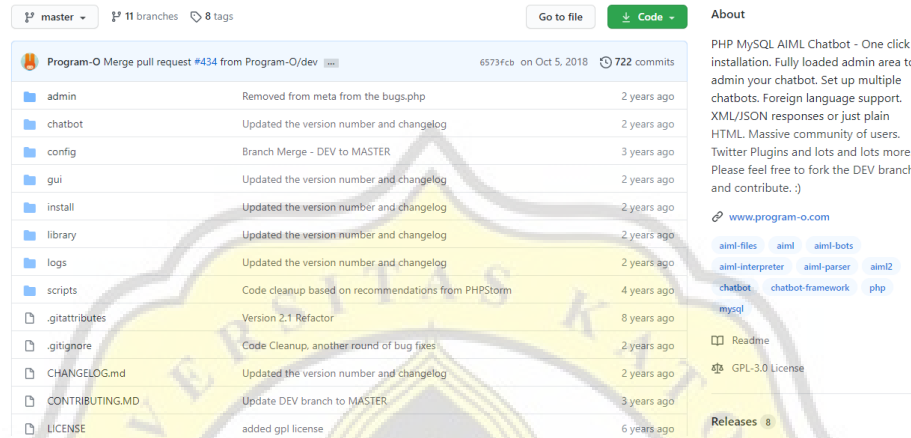


Gambar 4.31 laman Statistik Pengunjung

Hasil dari perancangan aplikasi di atas akan terlihat seperti pada gambar 4.31 di atas. Menunjukkan bahwa pengguna dapat melihat aplikasi Virtual Assistant dari arah depan, kanan, kiri, dan belakang, sehingga memberikan kesan hologram pada aplikasi Virtual Assistant tersebut.

b. Chatbot

Gambar 4.32 adalah website untuk mendownload Program-O
<https://github.com/Program-O/Program-O>.



Gambar 4.32 Download Chatbot

Langkah awal dalam pembuatan *chatbot* adalah dengan mengunduh sebuah program *open-source* bernama Program-O yang kemudian akan diekstrak pada direktori yang sudah disiapkan yaitu folder bernama “Virtual Assistant”.

Pada gambar di bawah ini adalah file-file yang tergabung di dalam Program-O.

Name	Date modified	Type	Size
admin	10/5/2018 4:48 AM	File folder	
chatbot	10/5/2018 4:48 AM	File folder	
config	10/5/2018 4:48 AM	File folder	
gui	10/5/2018 4:48 AM	File folder	
install	10/5/2018 4:48 AM	File folder	
library	10/5/2018 4:48 AM	File folder	
logs	10/5/2018 4:48 AM	File folder	
scripts	10/5/2018 4:48 AM	File folder	
.gitattributes	10/5/2018 4:48 AM	Text Document	1 KB
.gitignore	10/5/2018 4:48 AM	Text Document	3 KB
aiml tag list	10/5/2018 4:48 AM	Text Document	1 KB
CHANGELOG.md	10/5/2018 4:48 AM	MD File	30 KB
CONTRIBUTING.MD	10/5/2018 4:48 AM	MD File	1 KB
favicon	10/5/2018 4:48 AM	Icon	2 KB
fresh_install_README	10/5/2018 4:48 AM	Text Document	3 KB
getbots	10/5/2018 4:48 AM	PHP File	2 KB
index	10/5/2018 4:48 AM	PHP File	2 KB
LICENSE	10/5/2018 4:48 AM	File	32 KB
README.md	10/5/2018 4:48 AM	MD File	4 KB
robots	10/5/2018 4:48 AM	Text Document	1 KB
style	10/5/2018 4:48 AM	Cascading Style S...	1 KB
TODO	10/5/2018 4:48 AM	File	2 KB
version	10/5/2018 4:48 AM	Text Document	1 KB

Gambar 4.33 Program-O yang sudah diekstrak ke dalam folder Virtual Assistant

Gambar 4.33 menunjukkan aplikasi Program-O yang sudah diekstrak ke dalam direktori “Virtual Assistant” dengan sub-folder “chatbot”.

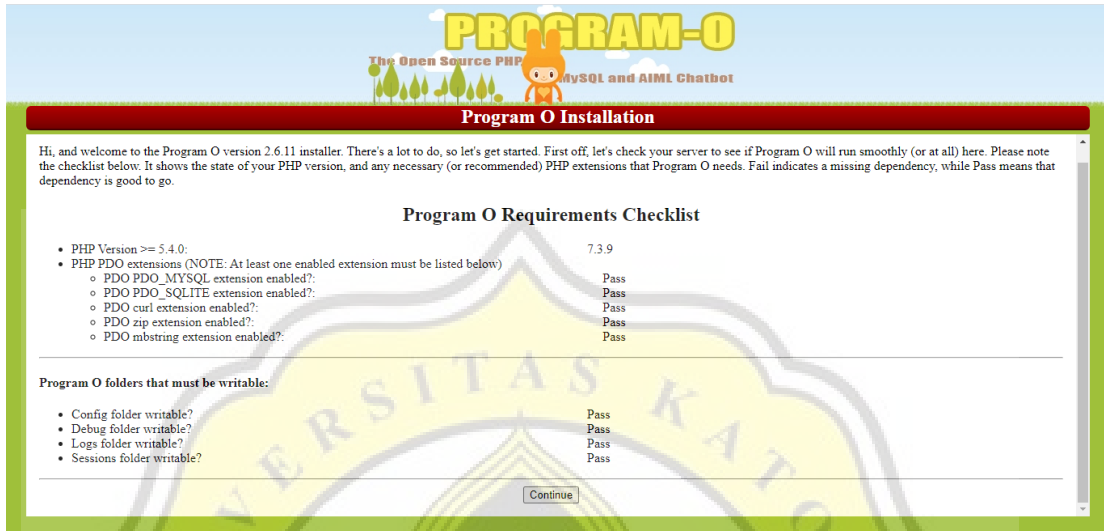
Gambar 4.34 adalah halaman yang ditampilkan jika anda pertama kali membuka Program-O melalui localhost.

Program O exists, but is not installed. [Install Program O](#)

Gambar 4.34 Program-O yang sudah dijalankan

Kemudian aplikasi Program-O dijalankan pada host <http://www.lib.unika.com/VirtualAssistant/chatbot> untuk memulai proses instalasi.

Pada gambar 4.35 Program-O melakukan pengecekan sistem pada komputer anda.



Gambar 4.35 Analisa Program-O

Program-O akan menganalisa apakah sistem yang dimiliki sudah memenuhi kategori dari Program-O tersebut. Jika semua kebutuhan dikatakan *Pass* maka proses dapat dilanjutkan.

Pada gambar di bawah ini adalah tampilan Program-O untuk penamaan dan menetapkan database

Gambar 4.36 Form Keterangan Bot

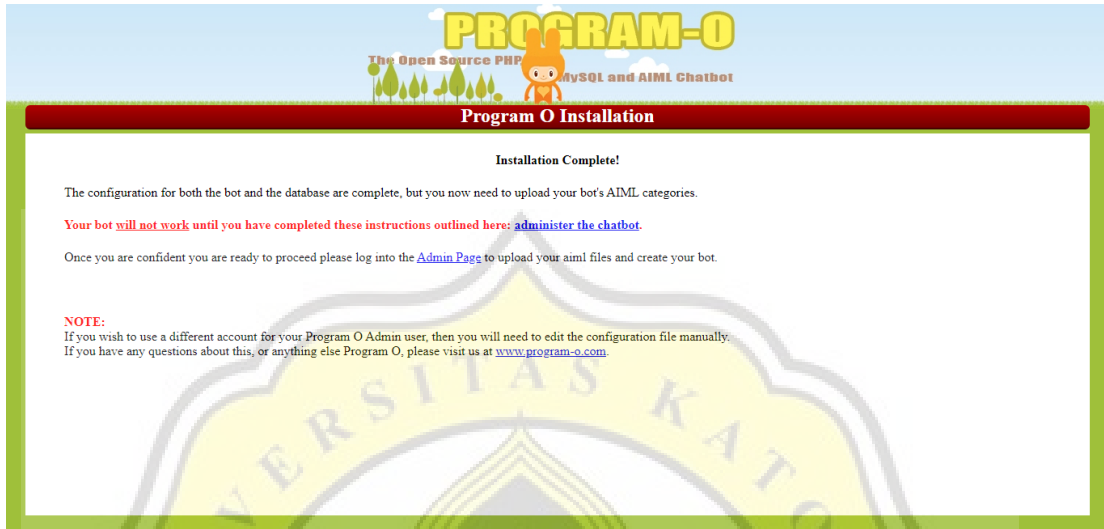
Setelah sistem yang dibutuhkan sudah terpenuhi maka Program O akan meminta keterangan admin, bot, dan database.

Gambar di bawah ini adalah tampilan dari halaman database MySQL.

Gambar 4.37 Membuat Database untuk Virtual Assistant

Program-O akan meminta database untuk menyimpan setiap data konfigurasi dan percakapan yang bot lakukan.

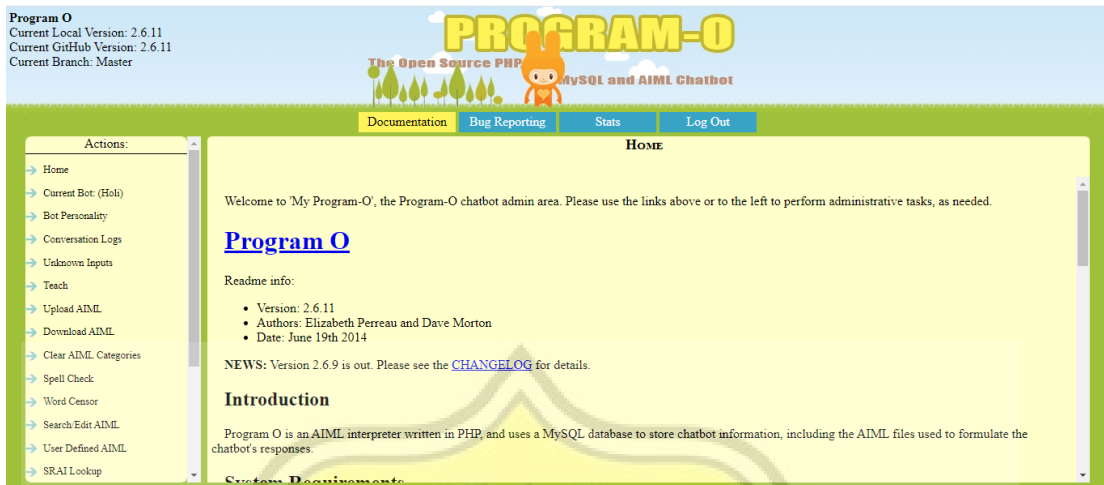
Gambar 4.38 adalah halaman yang akan anda dapatkan jika installasi Program-O berhasil.



Gambar 4.38 Intallasi Selesai

Setelah form instalasi sudah terpenuhi maka proses instalasi Program-O telah selesai. Kemudian tekan link “Admin Page” untuk login ke dashboard admin dari Bot yang sudah dibuat.

Gambar 4.39 adalah dashboard admin yaitu halaman utama dari Program-O.



Gambar 4.39 Dashboard Admin

Seperti pada gambar di atas dashboard admin dari Program-O yang sudah berhasil terinstall. Langkah berikutnya adalah dengan mengunggah AIML agar Bot dapat berkomunikasi dengan pengguna.

Pada gambar di bawah ini adalah format dari penulisan bahasa pemrograman

<AIML>.

```
<aiml version = '1.0.1' encoding = 'UTF-8'?>
  <category>
    <pattern>*</pattern>
    <template>
      Maaf Buku Yang Anda Cari Tidak Ada Atau Salah Dalam Pengetikan
    </template>
  </category>
  <category>
    <pattern>Manajemen : Jilid 1</pattern>
    <template>
      Buku Manajemen : Jilid 1 Yang Anda Cari Terletak di 658 STO M Lantai 3
    </template>
  </category>
```

Gambar 4.40 AIML

AIML memiliki bentuk yang menyerupai XML, dalam AIML terdiri dari <category> yang menampung informasi, <pattern> adalah input oleh pengguna, dan <template> adalah respon yang akan ditampilkan oleh bot. Untuk membuat sebuah percakapan yang besar akan membutuhkan banyak waktu dan tenaga, oleh sebab itu penulis merancang sebuah program untuk mengubah data dari database menjadi *script* AIML.

Pada gambar 4.41 adalah kode dari program untuk mengubah database, menjadi kode AIML.

```
<?php
include "config.php";

if($conn->connect_error){
    echo 'Connection Failed: '.$conn->connect_error;
}else{
    $sql="SELECT DISTINCT biblio.title,biblio.biblio_id,item.call_number, location.location_name
    FROM biblio
    INNER JOIN item
    INNER JOIN location
    ON biblio.biblio_id=item.biblio_id AND item.location_id = location.location_id;";

    $res=$conn->query($sql);
    $array = array();
    echo "<aiml version = '1.0.1' encoding = 'UTF-8'?>";
    while($row=$res->fetch_assoc()){
        if(!in_array($row["title"], $array)){
            array_push($array, $row["title"]);
            echo "
                <category>
                <pattern>".$row["title"]."</pattern>
                <template>
                Buku ".$row["title"]." Yang Anda Cari Terletak di ".$row["call_number"]." ".$row["location_name"]."<br>
                </template>
                </category>";
        }
        $arr_title = explode(' ', $row["title"]);
        $kata_depan = $arr_title[0];
        if(!in_array($kata_depan, $array)){
            array_push($array, $kata_depan);
            echo "
                <category>
                <pattern>".$kata_depan."</pattern>
                <template>
                Buku ".$kata_depan." Yang Anda Cari Terletak di ".$row["call_number"]." ".$row["location_name"]."<br>
                </template>
                </category>";
        }
        echo "Buku ".$row["title"]." Yang Anda Cari Terletak di ".$row["call_number"]." ".$row["location_name"]."<br>";
    }
    echo "</aiml>";
}
```

Gambar 4.41 AIML converter

Dengan mengubah data dari database menjadi AIML membuat proses pembuatan program Virtual Assistant Hologram Library menjadi lebih cepat. Proses pengubahan dari data menjadi AIML adalah dengan memilih data dari database, lalu menggunakan *echo* untuk menampilkan output pada layar yang disesuaikan dengan bentuk AIML seperti pada gambar 4.42.

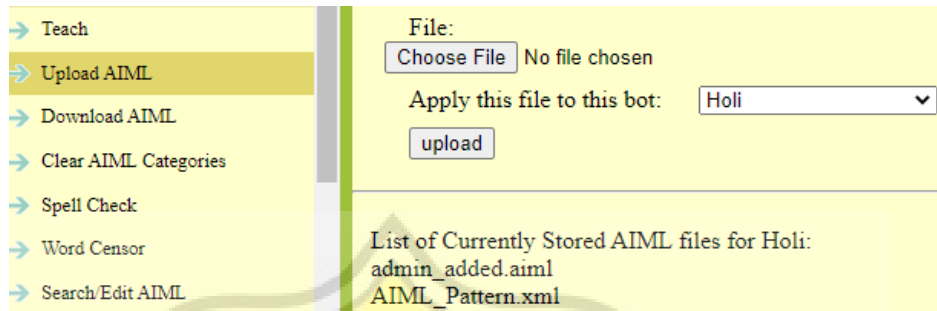
Gambar di bawah ini adalah hasil daftar buku dari database yang diubah menjadi kode AIML.

```
<aiml version = '1.0.1' encoding = 'UTF-8'?>
  <category>
    <pattern>Manajemen : Jilid 1</pattern>
    <template>
      Buku Manajemen : Jilid 1 Yang Anda Cari Terletak di 658 Sto m Lantai 3<br>
    </template>
  </category>
  <category>
    <pattern>Manajemen</pattern>
    <template>
      Buku Manajemen Yang Anda Cari Terletak di 658 Sto m Lantai 3<br>
    </template>
  </category>
  <category>
    <pattern>Management Produksi 1 Ed.3</pattern>
    <template>
      Buku Management Produksi 1 Ed.3 Yang Anda Cari Terletak di 658.5 Ahy m Lantai 3<br>
    </template>
  </category>
  <category>
    <pattern>Management</pattern>
    <template>
      Buku Management Yang Anda Cari Terletak di 658.5 Ahy m Lantai 3<br>
    </template>
  </category>
  <category>
    <pattern>Future Shock</pattern>
    <template>
      Buku Future Shock Yang Anda Cari Terletak di RB 303.4 Tof f Lantai 2<br>
    </template>
  </category>
  <category>
    <pattern>Future</pattern>
    <template>
      Buku Future Yang Anda Cari Terletak di RB 303.4 Tof f Lantai 2<br>
    </template>
  </category>
  <category>
    <pattern>Principles Of Pavement Design</pattern>
    <template>
      Buku Principles Of Pavement Design Yang Anda Cari Terletak di 625.8 yod p Lantai 3<br>
    </template>
  </category>
</aiml>
```

Gambar 4.42 Hasil dari pengkonversian ke AIML

Gambar 4.42 adalah hasil dari data yang sudah diubah ke AIML. Langkah selanjutnya adalah mengunggah file AIML ke dalam program O.

Gambar 4.43 adalah halaman untuk mengunggah file AIML yang terdapat pada dashboard admin Program-O.



Gambar 4.43 upload AIML

Mengunggah file AIML ke dalam program O semudah hanya dengan memilih file dari direktori, kemudian menentukan bot, dan klik *upload*. Maka file AIML akan otomatis tersimpan dalam program O sebagai percakapan.

Gambar di bawah ini adalah kode untuk API dari Program-O, sehingga dapat memanggil format chatbot cukup dengan menambahkan *conversation_start.php*.

```
3
4
5 * http://lib.unika.ac.id/VirtualAssistant/
6 * PROGRAM O
7 * Version: 2.1.3
8 * FILE: gui/plain/index.php
9 * AUTHOR: Jason Canggayuda
10 * DATE: 13 January 2019
11 * DETAILS: Hologram Library Virtual Assistant
12
13 $display = "";
14 $thisFile = __FILE__;
15 $test = "test123";
16
17 if (!file_exists('../config/global_config.php'))
18 {
19     header('Location: ../install/install_programo.php');
20 }
21
22 /** @noinspection PhpIncludeInspection */
23 require_once('../config/global_config.php');
24 session_name('ProgramO');
25 $debug_div = '';
26 $hideSP = '';
27 $resizeResponseDiv = '';
28 $clearButton = '';
29 $get_vars = (empty($_GET)) ? filter_input_array(INPUT_GET) : array();
30 $post_vars = (empty($_POST)) ? filter_input_array(INPUT_POST) : array();
31 $form_vars = array_merge($post_vars, $get_vars); // POST overrides and overwrites GET
32 if (empty($form_vars)) require_once('../chatbot/conversation_start.php');
33 $bot_id = (empty($form_vars['bot_id'])) ? $form_vars['bot_id'] : 1;
34 $say = (empty($form_vars['say'])) ? $form_vars['say'] : '';
35 $convo_id = (isset($form_vars['convo_id'])) ? $form_vars['convo_id'] : md5(time());
36 $format = (empty($form_vars['format'])) ? strtolower($form_vars['format']) : 'html';
37
38 if (ERROR_DEBUGGING)
39 {
40     $convo_id = (isset($form_vars['convo_id'])) ? $form_vars['convo_id'] : 'DEBUG'; // Hard-code the convo_id during debugging
41     $debug_src = (empty($form_vars) && file_exists(_DEBUG_PATH_ . "{$convo_id}.txt")) ? _DEBUG_URL_ . "reader.php?file={$convo_id}.txt" : '';
42     $debug_div = <<<endDebugDiv
43
44 <iframe id="debugDiv" src="$debug_src" frameborder="0">
45 endDebugDiv;
46 $hideSP = 'display: none;';
47 $resizeResponseDiv = 'max-height: 200px;';
```

Gambar 4.44 Chatbot API

Setelah proses pengunggahan file AIML ke dalam program O telah berhasil, selanjutnya adalah menggunakan API dari program O untuk mengirimkan *input* ke *database* yang kemudian dikirimkan kembali dalam berupa *response* (tanggapan balik). Pada gambar 4 44 di atas adalah bentuk dari API program O.

Pada gambar di bawah ini adalah *header* dari halaman pencarian buku.

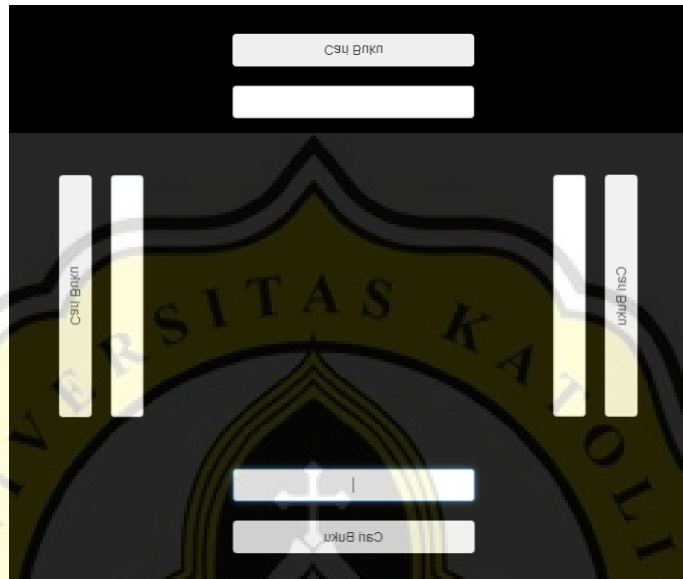
```
58 <!DOCTYPE html>
59 <html>
60 <head>
61   <meta charset="UTF-8">
62   <title>Hologram Library</title>
63 </head>
64 <link rel="stylesheet" href="../../css/style.css">
65 <!-- Latest compiled and minified CSS -->
66 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
67
68 <!-- jQuery library -->
69 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
70
71 <!-- Latest compiled JavaScript -->
72 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
73 <link rel="shortcut icon" href="">
74
75 <script type="text/javascript">
76   document.onkeydown = function(){
77     if(window.event && window.event.keyCode == 45)
78     {
79       window.location.href = "http://lib.unika.ac.id/VirtualAssistant/"
80     }
81   }
```

Gambar 4.45 Header Layar Pencarian Buku

Selanjutnya Header untuk layar pencarian buku dirancang. Pada gambar 4.45 adalah header untuk layar pencarian buku, pada baris ke 75 adalah *script javascript* yang berguna untuk memindahkan halaman ke halaman utama yaitu layar statistik pengunjung apabila layar pencarian buku tidak ada pergerakan (*idle*) selama 45 detik.

d. Layar Pencarian Buku

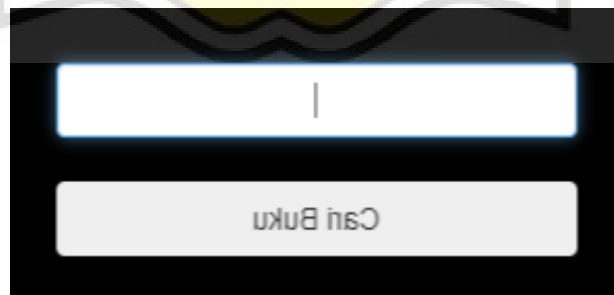
Gambar 4.46 adalah tampilan secara keseluruhan dari halaman pencarian buku.



gambar 4 .46 Tampilan pencarian buku

Pada gambar 4.46 terlihat tampilan untuk pencarian buku, secara keseluruhan memiliki metode yang sama dengan layar statistik, hanya saja yang menjadi pembeda adalah *input* yang terkirim. Pada layar pencarian buku *input* dari pengguna akan terkirim ke program API Program O yang kemudian jawaban dikelola dan dikirimkan kembali kepada pengguna.

Gambar di bawah ini adalah *form input* dari halaman pencarian buku.



gambar 4 .47 form input layar pencarian buku

Sebelum jawaban berupa *Virtual Assistant* ditampilkan *input* dari pengguna akan dikelola terlebih dahulu, jika buku yang dicari berupa judul atau nama pengarang atau topik yang memiliki hasil lebih dari 1 (satu) maka pengguna akan diarahkan ke layar pilih buku.

Gambar 4.48 adalah kode dari halaman pencarian buku.

```
<div class="div-6">
  <form name="chatform2" method="post" action="index.php#end"
    onsubmit="if(document.getElementById('say').value == '') return false;">
    <div id="input">
      <br>
      <input type="text" name="say2" id="say2" class="center form-control" autocomplete="off" onkeyup="ketikSemua()" size="70"/>
      <input type="submit" name="submit" class="input-button btn btn-secondary" value="Cari Buku"/>
      <input type="hidden" name="format" id="format2" value="html"/>
    </div>
  </form>
</div>
```

Gambar 4.48 Kode form pencarian buku

Pada gambar di atas adalah bentuk dari sebuah *form input* layar pencarian buku. Secara keseluruhan memiliki metode yang sama dengan statistik pengunjung.

Pada gambar 4.49 adalah kode Javascript untuk merespon *input* dari pengguna.

```
87 function tampilVideo()
88 {
89     if(document.getElementById('say').value == '');
90
91     var botsay_text = document.getElementById("botsay_text");
92     var video1 = document.getElementById("video1");
93
94     var chatform1 = document.getElementById("chatform1");
95
96     chatform1.style.display = "none";
97     video1.style.display = "block";
98     video1.play();
99     return false;
100 }
101
102 function ketikSemua(){
103     // alert("Terpanggil");
104     var say = document.getElementById("say").value;
105     var say2 = document.getElementById("say2");
106     var say3 = document.getElementById("say3");
107     var say4 = document.getElementById("say4");
108
109     say2.value = say;
110     say3.value = say;
111     say4.value = say;
112 }
```

Gambar 4.49 Kode untuk Menampilkan Video dan pengisian otomatis

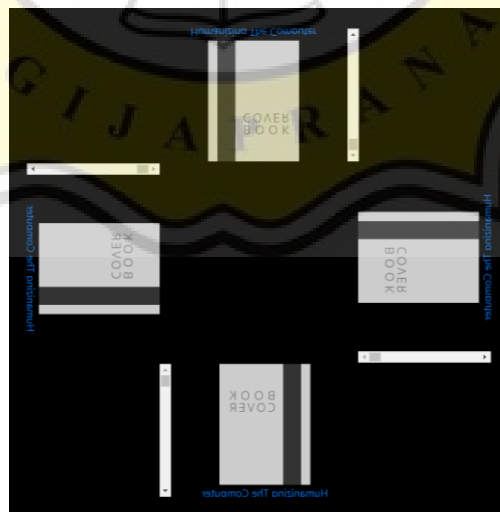
Function javascript seperti gambar di atas adalah *script* untuk mengirimkan *input* dari pengguna ke program O yang kemudian dilanjutkan ke layar berikutnya berisikan jawaban dari *chatbot*. Serta untuk *function* “ketikSemua” berfungsi agar setiap *form* yang terdapat pada layar hologram akan terisi secara otomatis berdasarkan satu *form* yang diisi oleh pengguna.

```
function say(m) {  
  var msg = new SpeechSynthesisUtterance();  
  var voices = window.speechSynthesis.getVoices();  
  msg.voice = voices[1];  
  msg.voiceURI = "native";  
  msg.volume = 1;  
  msg.rate = 1;  
  msg.pitch = 1;  
  msg.text = m;  
  msg.lang = 'id-tl';  
  speechSynthesis.speak(msg);  
}
```

Gambar 4.50 Script Text to Speech

Gambar 4.50 menunjukkan sebuah *function* untuk membaca teks menjadi suara, dalam layar pencarian buku, suara yang keluar adalah nama buku, dan lokasi di mana buku berada.

Pada gambar di bawah ini terdapat tampilan halaman *Select Book* secara keseluruhan dari keempat sisi.



Gambar 4.51 Halaman untuk Memilih Buku

Seperti yang sudah dibahas sebelumnya pengguna akan diarahkan ke layar *Select Book* untuk memilih buku yang persis dicari oleh pengguna.

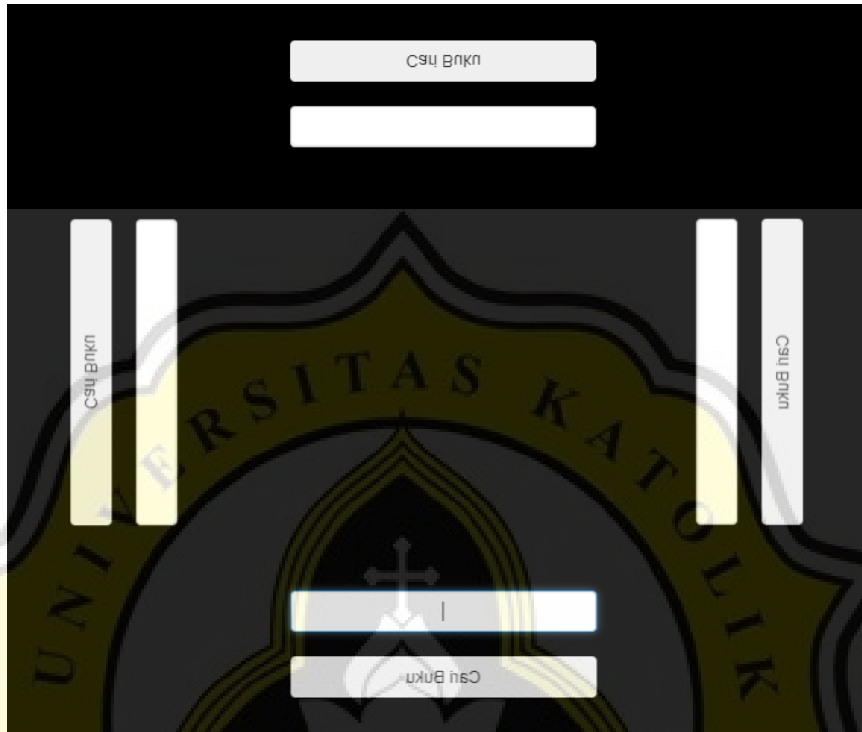
Gambar 4.52 adalah hasil dari *Virtual Assistant* untuk pencarian buku, yang ditampilkan secara terbalik agar gambar dipantulkan dan refleksi pada layar hologram akan menjadi tampilan yang sesungguhnya.



Gambar 4.52 Tampilan Virtual Assistant

Jika buku yang dicari telah terpilih maka *chatbot* akan mengirimkan *response* berupa teks dan suara seperti pada gambar 4.53. *Response* berupa judul buku dan lokasi di mana buku berada di dalam perpustakaan.

Gambar di bawah ini adalah tampilan yang akan pengguna dapatkan ketika telah menyelesaikan pencariannya.



Gambar 4.53 Kembali ke layar pencarian buku

Jika pencarian telah dilakukan, maka secara otomatis *Virtual Assistant* akan kembali ke layar pencarian buku.

d. Layar Idle

Pada gambar 4.54 adalah halaman *idle*.



Gambar 4.54 Layar Idle

Jika program *Virtual Assistant* tidak mendeteksi adanya aktivitas maka secara otomatis program akan mengarahkan pengguna ke layar idle seperti gambar di atas.

Gambar di bawah ini adalah kode untuk mengarahkan ke halaman *idle*.

```
172 <meta http-equiv="refresh" content="60;url=http://localhost/VirtualAssistant/idle/idle.php" />
173 <!-- <meta http-equiv="refresh" content="60;url=http://localhost/VirtualAssistant/idle/idle.php" /> -->
174 </body>
175
176 </html>
```

Gambar 4.55 Script Redirecting

Kode pada gambar 4.55 berguna untuk mengubah halaman utama ke halaman idle jika tidak ada aktivitas apapun selama 60 detik pada halaman utama.

Gambar 4.56 adalah kode dari halaman *idle*.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Hologram</title>
5 <link rel="stylesheet" href="../css/style.css">
6 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="
sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFc31SawIGgFAW/dAiS6JXm" crossorigin="anonymous">
7 </head>
8 <body class="bg-color" id="demo">
9 <div class="full-body">
10 <div class="div-content">
11 <!-- Atas -->
12 <div class="div-1">
13 <video width="320" height="240" autoplay loop>
14 <source src="new_project.mp4" type="video/mp4">
15 </video>
16 </div>
17 <!-- Kanan -->
18 <div class="div-2">
19 <video width="320" height="240" autoplay loop>
20 <source src="new_project.mp4" type="video/mp4">
21 </video>
22 </div>
23 <!-- Bawah -->
24 <div class="div-3">
25 <p id="notification">
26 <video width="320" height="240" autoplay loop>
27 <source src="new_project.mp4" type="video/mp4">
28 </video>
29 </div>
30 <!-- Kiri -->
31 <div class="div-4">
32 <video width="320" height="240" autoplay loop>
33 <source src="new_project.mp4" type="video/mp4">
34 </video>
35 </div>
36 </div>
37 </div>
38 </div>
```

Gambar 4.56 Layar Idle

Layar Idle berisikan video yang dimulai secara langsung dan *loop* pada gambar 4.56 menunjukkan bahwa program menjalankan 4 video dari berbagai arah.

Pada gambar di bawah ini terdapat kode untuk mengarahkan halaman *idle* kembali ke layar statistik/halaman utama jika terdapat aktivitas pada program tersebut.

```
<script>
  document.getElementById("demo").onkeypress = function() {myFunction()};

  function myFunction() {
    document.getElementById("demo").window.location.replace("http://localhost/VirtualAssistant/");
  }
</script>
```

Gambar 4.57 Kembali ke Laman Utama

Kemudian pada *Script idle* yang terakhir ada sebuah *javascript* yang berguna sebagai mengembalikan layar idle ke layar utama jika pengguna menekan tombol apapun pada keyboard

4.3 Rangkuman Hasil Wawancara

Berikut adalah hasil dari wawancara yang penulis lakukan kepada narasumber yakni empat staff beserta dengan kepala perpustakaan Universitas Katolik Soegijapranata yaitu:

1. **Kepala Perpustakaan:** Rikarda Ratih Saptaastuti (Kepala UPT Perpustakaan)
2. **Staff 1:** Agustin Hesti Pertiwi (Kepala TU)
3. **Staff 2:** Melania Adirati (Petugas Perpustakaan)
4. **Staff 3:** Kevin Christmanaya Adi (Kabag Pelayanan)
5. **Staff 4:** Norman (IT Perpustakaan)

No	Staff 1	Staff 2	Staff 3	Staff 4	Kepala Perpustakaan
1	Seperti yang dikatakan oleh bu Ratih, sangat membantu kami terutama dalam bidang pelayanan	Holi sudah sangat bermanfaat, sangat membantu di bidang pelayanan. Untuk Holi ini sendiri sangat membantu peran kami	Holi sangat-sangat bermanfaat, saya langsung berhadapan dengan HoLi, ini sangat membantu untuk saya tidak perlu memasukan nama dan nim	Fungsinya tidak kalah penting dengan sistem yang lain, tidak perlu repot, dan sangat membantu kami	Untuk saat ini cukup, walaupun Holi masih bisa di explore lagi. Saat ini statistik pengunjung, sapaan dan pencarian buku sudah bagus. Masih baru di Unika

		yang harus menyapa	mahasiswa, mereka hanya dengan membuka e-ktm dan scan.		
2	Sejauh ini sangat menjawab kebutuhan dari perpustakaan, mungkin kedepannya ada pengembangan lagi	Saya sangat setuju, jika HoLi menjawab kebutuhan perpustakaan masa kini seperti untuk pencarian buku	Untuk HoLi sudah menjawab kebutuhan yang ada di perpustakaan, namun perlu ada pengembangan lebih lagi karena tidak mungkin berhenti di sini	Kalo untuk HoLi sendiri sangat menjawab kebutuhan sekali. Untuk kedepannya mungkin bisa untuk diperbesar lagi hologramnya	Saat ini sudah dipakai mahasiswa untuk data statistik. Yang membuat beda adalah selain model layanan yang baru dengan konsep hologram, ini yang membuat perpustakaan unika menjadi berbeda, ketika ada kunjungan dari instansi lain, mereka cukup exited karna

					<p>ada sesuatu yang berbeda yang kita berikan ke mahasiswa.</p> <p>Mahasiswa generasi Alpha dan Z cukup senang karna mereka menemukan sesuatu yang baru terkait dengan perpustakaan karna menjadi mereka banget.</p>
3	<p>Saya baru pertama kali menggunakan aplikasi serupa HoLi</p>	<p>Belum pernah, ini pertama kali saya melihat aplikasi</p>	<p>Jika menggunakan memang baru pertama kali, namun saya pernah melihat</p>	<p>Belum pernah menemukan aplikasi yang sama</p>	<p>Belum pernah ditemukan aplikasi yang serupa</p>

		seperti ini, sangat bagus	di luar negeri tetapi di Indonesia baru kali ini apalagi menggunakan. Karna pertama kali saya menggunakan aplikasi serupa hanya di perpustakaan Unika Soegijapranata		
4	Awalnya kesulitan, tetapi setelah itu menjadi mudah	Bagi saya mudah	Bagi saya cukup mudah, kadang QR code Scanner kurang sensitif karena layar smartphone harus terang	Mudah, tetapi karena intensitas cahaya smartphone pengguna berbeda-beda sehingga harus lebih dijelaskan,	Mudah, karena mahasiswa tinggal masuk, buka e-ktm dan scan

				<p>terkadang juga QR Code E-KTM generate masih ada yang salah, terkadang bukan NIM yang keluar tetapi alamat Sintak. Tetapi bisa dibantu dengan input secara manual, jika QR Code ada yang bermasalah</p>	
5	Mudah dipahami	Bagi saya sudah cukup jelas	Alur untuk HoLi sudah jelas dan mudah dipahami	Ya, sudah jelas	Sosialisasinya cukup bagus kepada mahasiswa

6	Sama seperti ibu Ratih, HoLi menciptakan kesan yang membuat orang menjadi terkagum-kagum dengan keberadaannya	Pastinya sangat merubah kesan dan image menjadi lebih modern	Perpustakaan sudah menjadi lebih modern dan Juga membuat mahasiswa yang pertama kali datang menjadi kaget dan terkagum-kagum, karena mereka belum pernah menemukan aplikasi seperti ini.	Jika kembali ke jaman dulu, tidak ada yang seperti ini.	Pasti, merubah kesan dan image, memberikan kesan “ini perpustakaan aku banget lho”. Karna saya belum pernah menemukan di tempat yang lain
7	Iya, HoLi menjadi salah satu perwakilan dari perkembangan	Sudah sangat mewakili perkembangan teknologi, terlebih untuk Unika sebagai pioneer	Ya, sudah cukup mewakili	Sudah sangat mewakili	Iya, disatu sisi membuat kami senang karena menjadi sangat menarik, tetapi juga menjadi beban untuk

	teknologi di perpustakaan	pengguna HoLi itu sendiri			nanti ke depannya.
8	HoLi sesuai dengan kebutuhan promosi bagi perpustakaan saat ini	Iya, dan harapan kedepannya agar HoLi bisa digunakan di tempat-tempat yang lainnya	HoLi menjadi salah satu ujung tombaknya yang mewakili perpustakaan yang menjadi sumber informasi yang aktual dan up to date, jadi sekarang image perpustakaan sudah menjadi keren.	Di masa sekarang segalanya sudah harus serba otomatis, teknologi seperti HoLi ini sangat menunjang dan membuat pengunjung terkagum- kagum dan membuat orang beranggapan perpustakaan Unika ini maju.	Membantu aktivitas perpustakaan menjadi lebih efisien, dan juga HoLi menjadi <i>pioneer</i> bagi instansi yang lain ketika melakukan studi banding ke perpustakaan Unika

9	Menurut saya sangat disarankan sekali, karena sangat mempermudah mahasiswa, dan perpustakaan.	Bagi mahasiswa sangat disarankan, HoLi merupakan perkembangan teknologi yang sangat membantu pekerjaan kita. Untuk mahasiswa pun akan membantu mereka ke depannya	Sangat-sangat disarankan, karena lebih mudah digunakan, dan kedepannya akan ditambahkan fitur-fitur yang lain. Maka itu sangat disarankan bagi mahasiswa untuk menggunakan HoLi ini	Bisa, salah satunya pencarian buku.	Iya, dan memang saat ini holi ada dimanfaatkan di perpustakaan
10	Enjoy banget bagi saya	Enjoy dan sangat mengesankan bagi pengunjung perpustakaan,	Enjoy, gampang digunakan dan keren	Enjoy, karna sangat membantu	Ya enjoy banget, selain merubah image perpus menjadi lebih modern, dan juga

		<p>banyak yang mau datang ke perpustakaan karna banyak yang ingin tahu tentang HoLi</p>			<p>sebagai meningkatkan brand, dan sarana promosi juga. Kami menjadi punya keunikan sendiri dibandingkan perpustakaan yang lain</p>
11	Seharusnya iya	Menurut saya iya	Pastinya akan digunakan dan akan dikembangkan	Ya akan terus dipakai karena sudah menjadi iconic di Unika Soegijapranata . Jika HoLi dibuang maka akan menjadi kesalahan, karena sudah tidak ada ciri khas lagi.	Harus ya, dilain sisi karena investasi yang banyak dan proses pembuatan yang tidak mudah, pastinya harus terus dipakai, dan tidak hanya berhenti di sini.

--	--	--	--	--	--

