

**ANALISIS SISTEM REGISTRASI PADA APLIKASI ONLINE  
TICKETING CGV BLITZ**

**LAPORAN PROYEK I**

**Diajukan untuk memenuhi kelulusan matakuliah Proyek  
I pada Program Studi DIV Teknik Informatika**

**O l e h :**

**Rayhan Prastya  
1.18.4.069**



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

**BANDUNG  
2019**

## ABSTRAK

Memesan tiket secara online ( *e-ticket* ) dapat mempermudah kita memesan tiket tanpa harus mengantri untuk membeli tiket. Salah satu aplikasi yang menyediakan fitur *e-ticket* di indonesia adalah blitz. Blitz adalah sebuah satu aplikasi *e-ticket* berbasis *website*. Sebelum masuk ke tahap pemesanan tiket kita harus melakukan registrasi terlebih dahulu, nantinya makalah ini akan membahas hasil analisis yang telah dilakukan terhadap tahap registrasi di aplikasi blitz.

Tahap registrasi ini menggunakan bahasa pemrograman *PHP*, dan *MySQL* sebagai tempat manajemen data pengguna. Dan menggunakan teknik *Diagram Flow Data* (DFD) untuk menganalisis cara kerja dari aplikasi tersebut. Analisis ini bertujuan agar nantinya dapat memahami bagaimana cara kerja sistem registrasi tersebut dan kedepanya dapat mengimplementasikan ilmu yang didapat dari analisis sisttem ini.

**kata Kunci :** *e-ticket*, registrasi, *PHP*, *MySQL*, analisis.

## **ABSTRACT**

*Ordering tickets online (e-ticket) can make it easier for us to buy tickets without having to queue to buy tickets. One application that provides the e-ticket feature in Indonesia is CGV blitz. CGV blitz is one of the website-based e-ticket applications. Before entering into the ticket booking list we must register first, this paper will discuss the results of the analysis that has been done on the registration in the CGV Blitz application.*

*This registration phase uses the PHP programming language, and MySQL as a place for managing user data. And use the Data Flow Diagram technique (DFD) to analyze how the application works. This analysis was made in order to be able to discuss how the registration system works and in the future it can be applied to the knowledge obtained from this system analysis.*

**Keywords:** *online ticketing, registration, PHP, MySQL, analysis.*

## KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa karena dengan rahmat, karunia, serta taufik dan hidayah-Nya kami dapat menyelesaikan laporan project 1 yang berjudul Analisis Sistem Registrasi Pada Aplikasi Online Ticketing CGV-blitz Menggunakan Metode *data flow diagram* (DFD) dengan baik meskipun banyak kekurangan didalamnya. Kami juga ingin menyampaikan ucapan terima kasih kepada

1. Syafrial Fachri Pane, S.T., M.T.I. selaku Pembimbing dalam penyusunan Laporan Proyek I ini;
2. Roni Andarsyah, ST., M.Kom. selaku Penguji dalam pengujian hasil Laporan Proyek I ini;
3. Rd. Nuraini, S.F., S.S., M.Hum. selaku Koordinator Proyek I Tahun Akademik 2018/2019;
4. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika Tahun Akademik 2018/2019;

Kami berharap Laporan Proyek 1 ini dapat berguna dalam rangka menambah wawasan serta pengetahuan kita. Kami juga menyadari sepenuhnya bahwa di dalam Laporan ini terdapat kekurangan dan jauh dari kata sempurna. Oleh sebab itu, kami berharap adanya kritik, saran dan usulan demi perbaikan Laporan yang telah kami buat di masa yang akan datang, mengingat tidak ada sesuatu yang sempurna tanpa adanya saran yang membangun. Semoga Laporan ini dapat dipahami bagi siapa pun yang membacanya. Sekiranya Laporan yang telah disusun ini dapat berguna bagi kami maupun orang yang membacanya. Sebelumnya kami mohon maaf apabila terdapat kesalahan kata-kata yang kurang berkenan dan kami memohon kritik dan saran yang membangun demi perbaikan di masa depan.

Bandung, 8 Agustus 2019

Penulis

# DAFTAR ISI

<b>ABSTRAK</b>	<b>i</b>
<b><i>ABSTRACT</i></b>	<b>ii</b>
<b>KATA PENGANTAR</b>	<b>iii</b>
<b>DAFTAR ISI</b>	<b>iv</b>
<b>DAFTAR GAMBAR</b>	<b>vi</b>
<b>DAFTAR TABEL</b>	<b>vii</b>
<b>DAFTAR SIMBOL</b>	<b>viii</b>
<b>I PENDAHULUAN</b>	<b>I-2</b>
1.1 Latar Belakang . . . . .	I-2
1.2 Identifikasi Masalah . . . . .	I-3
1.3 Tujuan Dan Manfaat . . . . .	I-3
1.4 Ruang Lingkup . . . . .	I-4
1.5 Sistematika Penulisan . . . . .	I-4
<b>II LANDASAN TEORI</b>	<b>II-6</b>
2.1 Teori Umum . . . . .	II-6
2.1.1 <i>e-ticket</i> . . . . .	II-6
2.1.2 <i>Website</i> . . . . .	II-6
2.1.3 <i>Analisis</i> . . . . .	II-6
2.1.4 <i>PHP</i> . . . . .	II-6
2.1.5 <i>MySQL</i> . . . . .	II-6
2.1.6 <i>Flowchart</i> . . . . .	II-7
2.1.7 <i>Data Flow Diagram</i> . . . . .	II-7

2.1.8	API . . . . .	II-7
2.1.9	Google . . . . .	II-7
<b>III ANALISIS DAN PERANCANGAN</b>		<b>III-9</b>
3.1	Analisis Sistem . . . . .	III-9
3.1.1	Analisis Sistem Berjalan . . . . .	III-9
3.1.1.1	Analisis Prosedur ( <i>Flowmap</i> ) . . . . .	III-9
3.1.1.2	Analisis Dokumen Yang Digunakan . . . . .	III-11
3.1.2	Analisis Sistem Yang Akan Dibangun . . . . .	III-11
3.1.2.1	Analisis Kebutuhan Aplikasi . . . . .	III-13
3.1.2.2	Analisis Kebutuhan Perangkat Lunak Dan Perangkat Keras . . . . .	III-13
3.2	Perancangan . . . . .	III-15
3.2.1	<i>Context Diagram</i> . . . . .	III-15
3.2.2	<i>Data Flow Diagram</i> . . . . .	III-15
3.2.2.1	<i>DFD Level 0</i> . . . . .	III-16
3.2.3	Spesifikasi Proses ( <i>Process Specification</i> ) . . . . .	III-17
3.2.3.1	Spesifikasi Proses ( <i>Process Specification</i> ) DFD Level 0 . . . . .	III-17
3.2.4	Kamus Alir Data( <i>Data Flow Dictionary</i> ) . . . . .	III-17
3.2.5	Perancangan <i>Database</i> . . . . .	III-18
3.2.5.1	<i>Conceptual Data Model</i> . . . . .	III-18
3.2.5.2	<i>Physical Data Model</i> . . . . .	III-18
3.2.5.3	<i>Kamus Data Tabel(Database)</i> . . . . .	III-18
3.2.6	Struktur Menu . . . . .	III-20
3.2.7	Perancangan Antar Muka . . . . .	III-20
3.2.7.1	<i>Index</i> . . . . .	III-20
3.2.7.2	<i>Config</i> . . . . .	III-22
3.2.7.3	<i>Callback</i> . . . . .	III-22
3.2.7.4	<i>Login</i> . . . . .	III-23
3.2.7.5	<i>Dbconfig</i> . . . . .	III-25
3.2.7.6	<i>Function</i> . . . . .	III-26
3.2.7.7	Pembuatan <i>Database</i> . . . . .	III-27

<b>IV IMPLEMENTASI DAN PENGUJIAN</b>	<b>IV-28</b>
4.1 Lingkungan Implementasi . . . . .	IV-28
4.1.1 Perangkat Keras . . . . .	IV-28
4.1.2 Perangkat Lunak . . . . .	IV-28
4.2 Pembahasan Hasil Implementasi . . . . .	IV-28
4.3 Pengujian . . . . .	IV-29
4.3.1 <i>Black Box</i> . . . . .	IV-29
<b>V KESIMPULAN DAN SARAN</b>	<b>V-31</b>
5.1 Kesimpulan . . . . .	V-31
5.2 Saran . . . . .	V-31
<b>Daftar Pustaka</b>	<b>V-32</b>

# DAFTAR GAMBAR





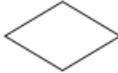






3.1	<i>Flowmap Sistem Yang Sedang Berjalan</i>	III-10
3.2	<i>Rancangan Flowmap</i>	III-12
3.3	<i>Context Diagram</i>	III-15
3.4	<i>DFDO</i>	III-16
3.5	<i>CDM</i>	III-18
3.6	<i>PDM</i>	III-18
3.7	<i>struktur menu</i>	III-20
3.8	<i>sub proses</i>	III-20
3.9	<i>Fungsi Index</i>	III-21
3.10	<i>Fungsi PHP Pada Index</i>	III-21
3.11	<i>HTML Pada Index</i>	III-22
3.12	<i>Fungsi PHP Pada Config</i>	III-22
3.13	<i>Fungsi PHP Pada Callback</i>	III-23
3.14	<i>Fungsi Pada Login</i>	III-24
3.15	<i>Fungsi PHP Pada Login</i>	III-24
3.16	<i>Fungsi HTML Pada Login</i>	III-25
3.17	<i>Fungsi PHP Pada Dbconfig</i>	III-25
3.18	<i>Fungsi PHP Pada Function</i>	III-26
3.19	<i>Pembuatan Database</i>	III-27
4.1	<i>Halaman Login</i>	IV-29
4.2	<i>Masuk Akun Google</i>	IV-29



# DAFTAR TABEL

3.1	Dokumen Yang Digunakan . . . . .	III-12
3.2	Deskripsi perangkat <i>server</i> . . . . .	III-14
3.3	Deskripsi perangkat <i>client</i> . . . . .	III-14
3.4	Deskripsi perangkat Lunak <i>Server</i> . . . . .	III-14
3.5	Deskripsi Perangkat Lunak <i>Client</i> . . . . .	III-15
3.6	<i>tabel spesifikasi DFD level 0</i> . . . . .	III-17
3.7	Kamus Aliran Data . . . . .	III-17
3.8	<i>User</i> . . . . .	III-19
3.9	<i>Captcha</i> . . . . .	III-19
3.10	<i>City</i> . . . . .	III-19
3.11	<i>Cinema</i> . . . . .	III-19
4.1	Deskripsi Perangkat Keras . . . . .	IV-28
4.2	Deskripsi perangkat Lunak <i>Client</i> . . . . .	IV-29
4.3	Blackbox . . . . .	IV-30
4.4	<i>Masuk Akun Google</i> . . . . .	IV-30

## DAFTAR SIMBOL

No.	Simbol	Nama	Keterangan
1		Terminal / Terminator	Menunjukkan awal atau akhir dari aliran proses.
2		Arrows	Menunjukkan arus data antar simbol / proses.
3		Data	Menunjukkan data yang menjadi input / output proses.
4		Process	Menunjukkan kegiatan proses dari operasi program komputer.
5		Decision	Menunjukkan pilihan yang akan dikerjakan atau keputusan yang harus dibuat dalam proses pengolahan data.
6		Preparation	Pemberian nilai awal suatu variable.
7		Connector (On-page connector)	Digunakan untuk penghubung dalam satu halaman.
8		Connector (Off-page connector)	Digunakan untuk penghubung berbeda halaman.
9		Document	Menunjukkan dokumen sebagai yang digunakan untuk merekam data terjadinya suatu transaksi.
10		Database	Menyimpan data berbasis database.
11		Manual Operation	Menunjukkan proses yang dikerjakan secara manual.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Di zaman yang serba modern dan serba online ini banyak perusahaan besar yang menggunakan kesempatan ini untuk meraup untung sebanyak-banyaknya dengan menggunakan teknik pemasaran melalui internet tetapi tetap mementingkan dan memudahkan para *customer* untuk memesan ataupun membeli sesuatu yang mereka butuhkan. Pemasaran di internet cenderung menembus berbagai rintangan batas bangsa dan tanpa aturan-aturan yang baku. Sedangkan pemasaran konvensional lebih banyak yang terlibat dibandingkan pemasaran lewat internet[12]. Sehingga menurut kami perusahaan dapat meraup keuntungan lebih besar dengan pemasaran melalui internet dibandingkan secara konvensional. Seperti perusahaan CGV-blitz yang ada di Indonesia. CGV-blitz ini merupakan salah satu perusahaan *E-ticket* berbasis *website* yang ada di Indonesia, *e-ticket* sendiri merupakan sebuah kemajuan dalam bidang teknologi karena kita tidak mengantri lagi untuk membeli tiket. Penerapan tiket elektronik merupakan langkah yang tepat sebagai salah satu upaya untuk mengefisiensi waktu dalam sebuah transaksi bisnis. Kelebihan dari aplikasi *e-tiket* bioskop online ini berguna sebagai media dalam pembelian tiket bioskop dengan media online tanpa harus antri di gedung bioskop untuk membeli tiket bioskop tersebut[5]. Penggunaan tiket elektronik telah berkembang pesat. Menurut *Airline Reporting Corporation (ARC)*, hampir (99 persen) dari semua maskapai tiket yang digunakan di Amerika Serikat (AS) pada 2013 diperoleh secara elektronik [2]. Dibandingkan membeli tiket secara manual yang akan sangat memakan waktu dan tidak efisien. Hal ini tentu saja membawa dampak kerugian karena waktu yang digunakan untuk mengantri dapat menjadi lebih efektif dengan melakukan hal-hal lain yang lebih bermanfaat[5].

Banyak pelanggan yang harus antri cukup lama untuk mendapatkan tiket, bahkan terkadang sudah mengantri pun belum tentu berhasil mendapatkan tiket. Namun sebelum kita bisa memesan tiket secara online kita harus memiliki akun terlebih dahulu, dan dalam langkah-langkah membuat akun, hal pertama yang harus dilakukan adalah registrasi. Di dalam laporan ini kita akan melakukan analisis terhadap proses registrasi tersebut. Dan dalam menganalisis proses registrasi ini

lebih dalam kita akan menggunakan cara analisis *Data Flow Diagram* (DFD). *Data Flow Diagram* (DFD) adalah alat visual untuk menggambarkan model logika dan mengekspresikan data transformasi dari input ke output dalam suatu sistem. Telah banyak digunakan kedua analisis kebutuhan dan analisis terstruktur dalam pengembangan perangkat lunak [13] . Diharapkan analisis ini nantinya dapat membantu kita memahami tentang cara kerja sistem registrasi tersebut, dan juga membuat tambahan fitur baru yaitu *Login* menggunakan Google+ *API*.

Dalam proses analisis registrasi akun CGV BLITZ ini kami memilih menggunakan bahasa pemrograman *PHP* dan *MySQL* karena kami merasa menggunakan bahasa pemrograman *PHP* dan *MySQL* lebih mudah dibandingkan menggunakan bahasa pemrograman yang lainnya. *PHP* adalah akronim dari *Hypertext Preprocessor*, yaitu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkan kembali ke web *browser* menjadi kode *HTML* [3]. *MySQL* merupakan database yang awalnya hanya berjalan pada sistem *Unix* dan *Linux*. Seiring berjalannya waktu dan banyaknya peminat yang menggunakan database ini, *MySQL* merilis versi yang dapat diinstal pada hampir semua platform, termasuk *Windows* [11] .

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang ada, maka dapat diidentifikasi menjadi beberapa masalah sebagai berikut:

1. Bagaimana proses registrasi pada akun CGV-Blitz ?
2. Bagaimana penggunaan bahasa pemrograman *PHP* dan *MySQL* dalam registrasi akun blitz

## 1.3 Tujuan Dan Manfaat

Tujuan dari di analisis sistem registrasi ini adalah sebagai berikut:

1. Melakukan analisis proses registrasi di CGV-Blitz menggunakan teknik *Data Flow Diagram* (DFD)
2. Melakukan analisis penerapan fungsi atau bahasa pemrograman *PHP* dan *MySQL* dalam registrasi akun CGV-Blitz

## 1.4 Ruang Lingkup

Kami membatasi ruang lingkup yang akan kami bahas sebagai berikut:

1. Proses pendaftaran dimulai, dari mengisi data diri hingga proses pendaftaran itu selesai dan masuk ke tampilan utama
2. Membahas fungsi-fungsi dasar bahasa program yang akan digunakan untuk membangun aplikasi tersebut

## 1.5 Sistematika Penulisan

Dalam sistematika penulisan ini menyangkut tentang materi-materi yang ada pada Laporan Proyek 1 ini, yang dibagi-bagi menjadi beberapa subab sebagai berikut:

### 1. PENDAHULUAN

Didalam bab ini menjelaskan secara umum tentang analisi yang akan dibahas pada proyek ini metode yang akan digunakan untuk menyelesaikan masalah pada analisis ini.

### 2. LANDASAN TEORI

Didalam bab ini berisikan tentang teori-teori yang bersumber dari Jurnal ataupun buku-buku yang berupa pengertian atau definisi yang berkaitan dengan Analisis yang dilakukan pada pembuatan Proyek 1 ini.

### 3. ANALISIS DAN PERANCANGAN

Didalam bab ini menjelaskan tentang function yang terdapat pada sistem registrasi online, dan yang akan dianalisis terdiri pembuatan *Data Flow Diagram* (DFD), dan bahasa pemrograman yang digunakan dalam aplikasi tersebut.

#### 4. IMPLEMENTASI DAN PENGUJIAN

Didalam bab ini berisi tentang pengujian dan hasil implementasi dari sistem yang akan dirancang.

#### 5. KESIMPULAN DAN SARAN

Didalam bab ini berisi kesimpulan dan Saran yang berkaitan dengan analisis laporan yang telah dibuat pada bab sebelumnya sehingga pembaca dimudahkan mengerti hasil dari laporan analisis ini.

# BAB II

## LANDASAN TEORI

### 2.1 Teori Umum

#### 2.1.1 *e-ticket*

E-Ticket adalah singkatan dari electronic ticket, atau dalam bahasa Indonesia diartikan tiket elektronik. Sehingga e-ticket adalah tiket yang wujudnya berbentuk elektronik [5].

#### 2.1.2 *Website*

*Website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah *domain* atau *subdomain*, yang tempatnya berada di dalam *World Wide Web* (WWW) di *internet* [12]

#### 2.1.3 *Analisis*

Analisis adalah suatu upaya penyelidikan untuk melihat, mengamati, mengetahui, menemukan, memahami, menelaah, mengklasifikasi, dan mendalami serta menginterpretasikan fenomena yang ada [1]

#### 2.1.4 *PHP*

*PHP* merupakan bahasa berbentuk *script* yang ditempatkan didalam *server* baru kemudian diproses. Kemudian hasil pemrosesan dikirim kepada *web browser client*. Bahasa pemrograman ini dirancang khusus untuk membentuk *web* dinamis. Artinya, pemrograman *PHP* dapat membentuk suatu tampilan berdasarkan permintaan terkini, misalnya halaman yang menampilkan daftar tamu. Halaman tersebut akan selalu mengalami perubahan mengikuti jumlah data tamu yang telah mengisi buku tamu [7]

#### 2.1.5 *MySQL*

*MySQL* adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). *MySQL* sebenarnya

merupakan turunan salah satu konsep utama dalam database sejak lama yaitu *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengoperasian database terutama untuk pemilihan/seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah dan secara otomatis. Keandalan suatu sistem *database* dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query My SQL* dapat sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase* [8]

### **2.1.6 Flowchart**

*Flowchart* merupakan bagan (*chart*) yang menunjukkan alir atau arus (*flow*) di dalam program atau prosedur *system* secara logika. *Flowchart* (bagan alir) merupakan gambaran dalam bentuk diagram alir dari *algoritma-algoritma* dalam suatu program, yang menyatakan arah alur program tersebut. [10]

### **2.1.7 Data Flow Diagram**

*Data Flow Diagram* (DFD) adalah salah satu yang tertua alat terstruktur tersedia untuk mendukung analisis sistem dan Desain. Nantinya DFD akan digunakan sebagai alat untuk menganalisa cara kerja yang ada di sistem tersesebut [9]

### **2.1.8 API**

*API* atau *Application Programming Interface* bukan hanya satu set *class* dan *method* atau fungsi dan *signature* yang sederhana. Akan tetapi *API*, yang bertujuan utama untuk mengatasi “*clueless*” dalam membangun *software* yang berukuran besar, berawal dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami. Secara sederhana dapat dipahami dengan membayangkan kekacauan yang akan timbul bila mengubah *database* atau skema *XML*. Perubahan ini dapat dipermudah dengan bantuan *API* [6].

### **2.1.9 Google**

Google adalah suatu mesin pencari yang sangat trend di zaman sekarang ini, google sangat banyak digunakan oleh manusia untuk membantu mencari informasi, baik itu dalam proses belajar mengajar, mengetahui berita, lowongan pekerjaan dan lain



sebagainya. Dimana penggunaan google sangat mudah dan kebanyakan *user* sangat suka menggunakan google. Seiring berkembangnya zaman google semakin canggih dan semakin banyak pengguna yang menggunakan google. Sehingga perpustakaan kurang diminati dalam pencarian informasi [4].

# BAB III

## ANALISIS DAN PERANCANGAN

### 3.1 Analisis Sistem

Analisis sistem dapat diidentifikasi sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi, dan kebutuhan yang diharapkan sehingga dapat menganalisis sistem yang berjalan. Pada bagian ini, hanya akan dibahas mengenai analisis pendaftaran online pemesan tiket bioskop melalui website CGV-blitz yang akan digambarkan dalam bentuk *flowmap* dan *Data Flow Diagram*.

#### 3.1.1 Analisis Sistem Berjalan

Gambaran tentang sistem yang saat ini sedang berjalan di *website* CGV-blitz pada bagian proses pendaftaran online. Kami menganalisis fungsi bahasa pemrograman dan cara kerja sistem yang terdapat dalam proses tersebut. Analisis sistem ini bertujuan untuk mengetahui bagaimana cara kerja *function* dan cara kerja sistem yang ada.

##### 3.1.1.1 Analisis Prosedur (*Flowmap*)

###### A. Analisis Sistem Pendaftaran Online

Analisis sistem yang sedang berjalan pada sistem pendaftaran online di *website* CGV-blitz, bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem dan *function-function* yang tersedia. Perancangan analisis sistem yang sedang berjalan yang dilakukan berdasarkan urutan kejadian tersebut dapat dibuat diagram aliran (*flowmap*) prosedur sistem pendaftaran online, berikut gambar 3.1 yang menjelaskan *flowmap* sistem yang sedang berjalan di blitz :

Berdasarkan *flowmap* pada gambar 3.1, berikut adalah rincian aliran dari *flowmap* tersebut

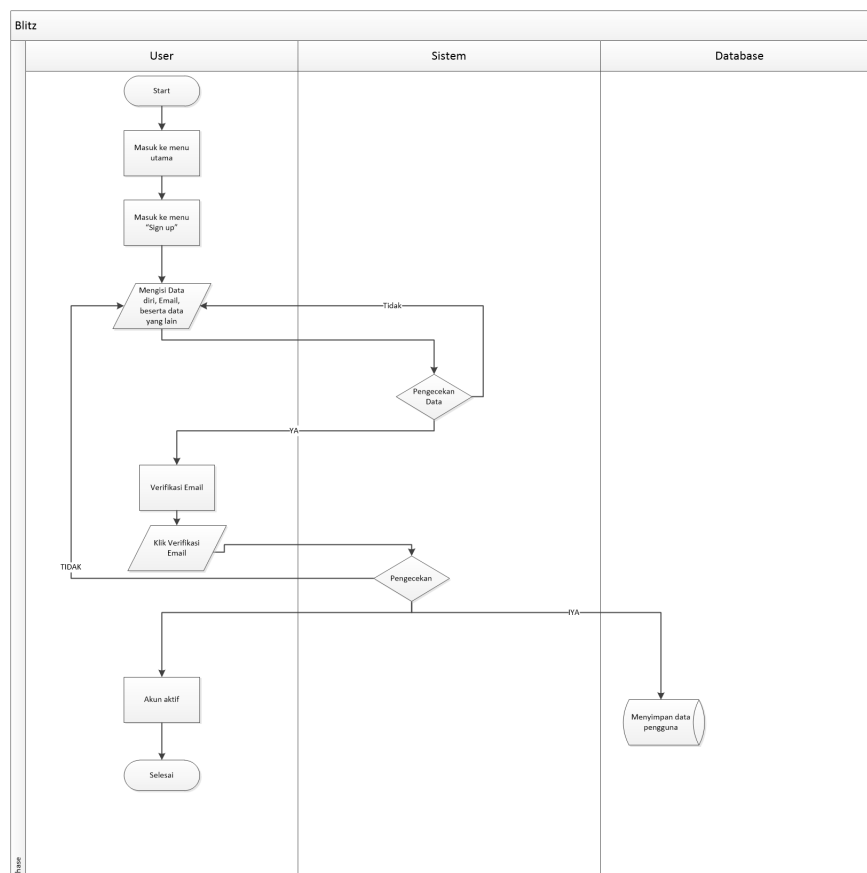


Figure 3.1: *Flowmap Sistem Yang Sedang Berjalan*

1. Start.
2. *User* akan dialihkan ke menu utama.
3. Lalu pengguna akan menekan tombol "*sign up*", dan akan tertera form registrasi pada halaman tersebut.
4. Pengguna akan mengisi data diri sesuai data yang user miliki.
5. Sistem akan melakukan pengecekan apakah data tersebut sudah valid atau belum, jika belum pengguna akan dialihkan kembali ke menu pengisian data diri dan melakukan pengisian ulang, namun jika data sudah benar maka sistem akan memproses data tersebut.
6. Sistem akan mengirim validasi *email* kepada pengguna.
7. Pengguna harus melakukan validasi kepada *email*-nya, jika tidak maka pengguna harus kembali dan melakukan pengisian data diri ulang, namun jika *email* sudah diverifikasi maka sistem akan kembali memproses verifikasi yang dilakukan oleh *user*.
8. Sistem akan melakukan penyimpanan data *user* kedalam *database* sistem.
9. Sistem akan melakukan aktivasi akun *user*
10. Akun *user* telah aktif
11. Selesai

#### **3.1.1.2 Analisis Dokumen Yang Digunakan**

Analisis dokumen merupakan penjelasan mengenai dokumen-dokumen yang digunakan dalam sistem pendaftaran *online*. Dalam tabel 3.1 merupakan analisa dokumen yang menjelaskan hal-hal yang akan dilakukan.

#### **3.1.2 Analisis Sistem Yang Akan Dibangun**

Analisis kebutuhan yang dimaksud disini berupa analisis *flowmap* mengenai sistem yang akan dibangun hanya meliputi sistem pendaftaran *online website* CGV-blitz.

Table 3.1: Dokumen Yang Digunakan

<b>Dibuat oleh</b>	CGV-BLITZ
<b>Dibuat untuk</b>	<i>Konsumen</i> (Masyarakat)
<b>Isi</b>	list syarat pengajuan pendaftaran online akun CGV-BLITZ
<b>Frekuensi</b>	Dibuat sesuai dengan kebutuhan bersyarat yang digunakan untuk melakukan pendaftaran online
<b>Tujuan</b>	Mempermudah proses membeli tiket bioskop online bagi masyarakat

#### A. Analisis sistem yang akan dibangun pada sistem pendaftaran *online*

Didalam proses pendaftaran *online* CGV-blitz ini, *user* harus memasukkan beberapa data, dan hal itu kurang efektif waktu. Jadi kami menambahkan fitur melakukan registrasi menggunakan akun Gmail yang dimiliki *user*.

Berikut gambar3.2 yang menjelaskan tentang *flowmap* yang akan dirancang.

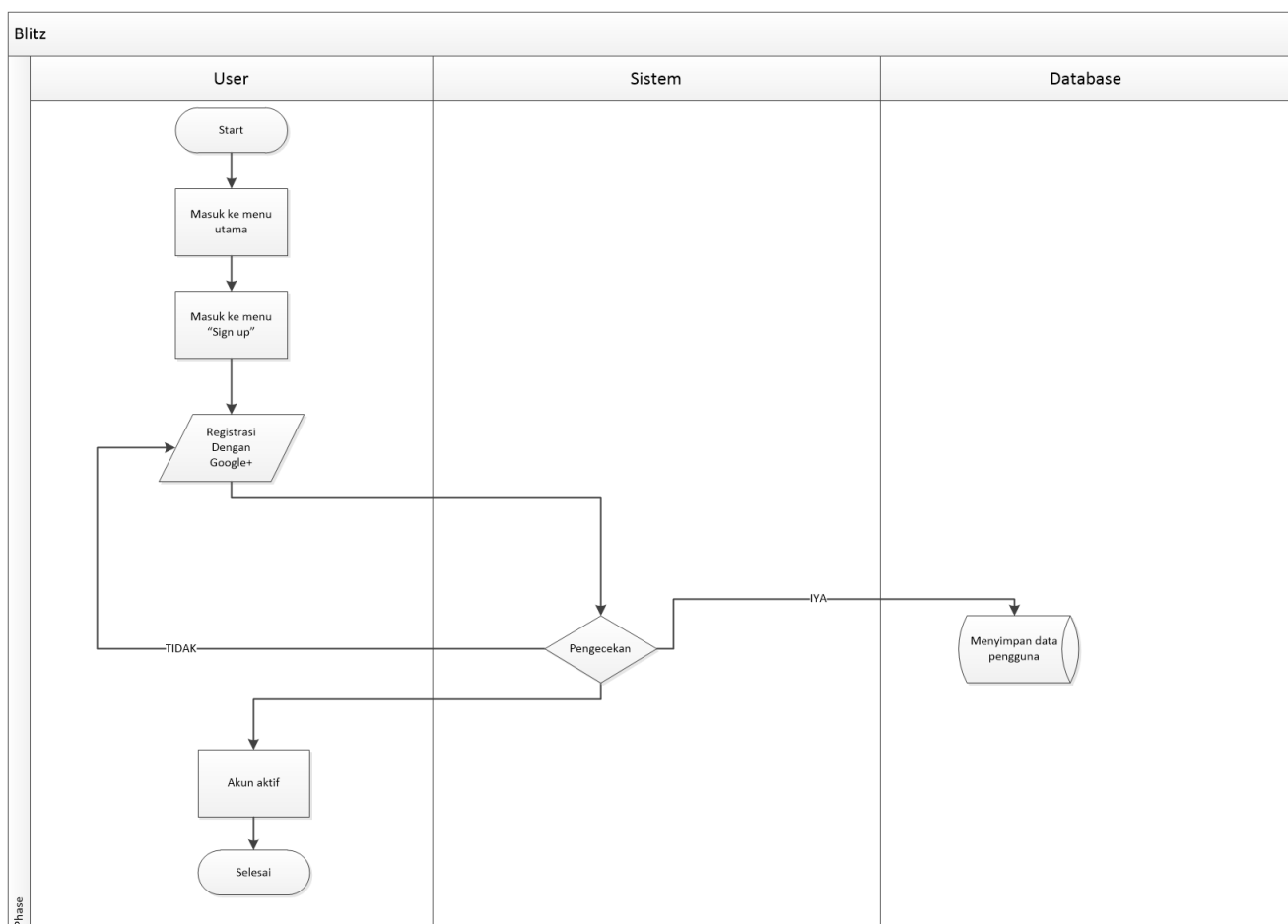


Figure 3.2: Rancangan *Flowmap*

Berdasarkan gambar *flowmap*3.2, berikut adalah rincian aliran *flowmap* yang akan dirancang nantinya

1. *Start*.
2. Pengguna akan dialihkan ke menu utama.
3. Lalu pengguna akan menekan tombol "*sign up*", dan akan tertera form registrasi pada halaman tersebut.
4. Registrasi menggunakan akun Google+
5. Sistem akan melakukan pengecekan apakah data tersebut sudah valid atau belum, jika belum maka pengguna akan diarahkan kembali ke menu registrasi, namun jika data sudah benar maka sistem akan memproses data tersebut.
6. Data *user* akan dimasukkan ke dalam *database*, dan akun *user* sudah aktif.
7. Selesai

#### **3.1.2.1 Analisis Kebutuhan Aplikasi**

Analisis kebutuhan aplikasi merupakan suatu kebutuhan yang berhubungan dengan kebutuhan sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan aplikasi yang akan dibuat yaitu pengelolaan data terdiri dari 2 proses sesuai dengan urutan sebagai berikut:

1. Pendaftaran Akun melalui Google+
2. Aktivasi Akun

#### **3.1.2.2 Analisis Kebutuhan Perangkat Lunak Dan Perangkat Keras**

Analisis kebutuhan perangkat lunak dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis pengguna/*User*. berikut tabel 3.2 ; 3.3 ; 3.4 ; 3.5 yang menjelaskan kebutuhan perangkat lunak dan perangkat keras yang nantinya akan digunakan.

**A. Kebutuhan Perangkat Keras**

**B. Kebutuhan Perangkat Lunak**

**C. Analisis Pengguna/*User***

Table 3.2: Deskripsi perangkat *server*

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Harddisk</i>	500 GB	Media untuk menyimpan data aplikasi yang dibuat
2	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan
3	<i>processor</i>	Intel® core™ i5-8250U CPU @ 1.60GHz (8 CPUs), 1.8Ghz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4	Infrastruktur Jaringan		Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses

Table 3.3: Deskripsi perangkat *client*

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Harddisk</i>	250 GB	Sebagai tempat penyimpanan data yang dibutuhkan, tetapi pada sisi <i>client</i> tidak diharuskan memiliki ketersediaan space yang besar
2	<i>Memory</i>	4 GB	Kecepatan <i>client</i> dalam mengakses system ini
3	<i>processor</i>	Intel® core™ i5-8250U CPU @ 1.60GHz (8 CPUs), 1.8Ghz	Untuk per halamanansi komputer
4	Infrastruktur Jaringan		<i>Server</i> dan <i>Host</i> alur proses dari titik awal proses sampai pada akhir proses

Table 3.4: Deskripsi perangkat Lunak *Server*

No	<i>Tools</i>	Fungsi	Keterangan
1	<i>Windows 10</i>	Sistem Operasi	-
2	<i>Xampp 1.7.3</i>	<i>server</i> Basis Data	-
3	<i>Photoshop CS</i>	desain antar muka	-
4	<i>PHP,HTML,CSS</i>	<i>Bahasa Program</i>	-
5	<i>sublime text</i>	software pendukung	-
6	<i>google chrome</i>	<i>browser</i>	-

Table 3.5: Deskripsi Perangkat Lunak *Client*

No	Tools	Fungsi	Keterangan
1	Windows 10	Sistem Operasi	-
2	Xampp 1.7.3	server Basis Data	-
3	Photoshop CS	desain antar muka	-
4	PHP,HTML,CSS	Bahasa Program	-
5	sublime text	software pendukung	-
6	google chrome	browser	-

Aplikasi yang akan dibuat ini hanya mencakup proses registrasi yang kelak nantinya akan digunakan oleh *user* untuk melakukan pendaftaran akun. Yang diantaranya melibatkan Pengguna (masyarakat yang akan mendaftar pada akun CGV-BLITZ).

## 3.2 Perancangan

### 3.2.1 Context Diagram

*Context diagram* berikut ini akan menggambarkan bagaimana proses keseluruhan dari sistem registrasi yang sedang berjalan di CGV blitz pada gambar 3.3 merupakan *context diagram* yang di analisis pada proses registrasi di aplikasi CGV Blitz

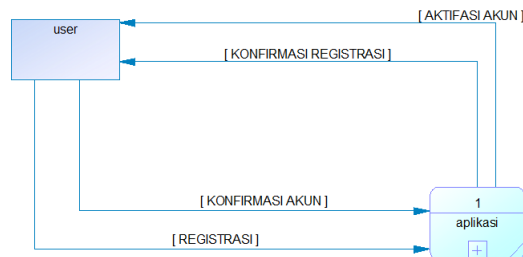


Figure 3.3: Context Diagram

### 3.2.2 Data Flow Diagram

*Data Flow Diagram* berikut akan memperinci pembahasan sistem registrasi yang sedang berjalan, yang nantinya dibagi menjadi beberapa proses. *Data Flow Diagram* ini nantinya hanya mencakup *DFD level 0*.



### 3.2.2.1 DFD Level 0

Seperti dijelaskan sebelumnya, *DFD level 0* ini merupakan penjabaran dari *Context diagram* pada sistem aplikasi ini. *DFD level 0* ini terdiri dari 2 (dua) proses sesuai dengan urutan sebagai berikut:

1. Proses registrasi
2. Aktivasi akun

Berikut pada gambar 3.4 merupakan *DFD Level 0* sistem registrasi pada aplikasi CGV-blitz.

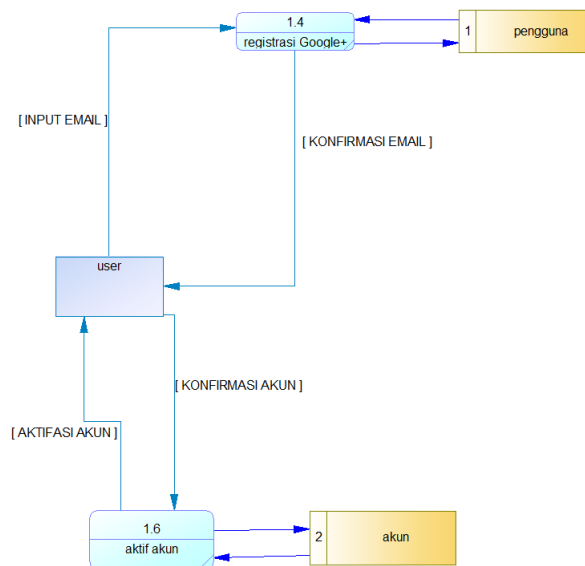


Figure 3.4: *DFD0*

Berdasarkan pada 3.4 *DFD Level 0* mencakup :

1. Hanya terdapat satu entitas utama yaitu *user*(pengguna aplikasi)
2. Terdapat 2 proses utama yang berjalan pada sistem, yaitu proses registrasi, proses aktivasi akun. Dimana setiap proses ini mewakili beberapa proses dalam sistem dengan rincian sebagai berikut
  - (a) Proses registrasi mencakup proses registrasi data diri *user* atau bisa menggunakan akun *email* pengguna untuk melakukan registrasi.
  - (b) Aktivasi akun meliputi pengaktifan akun pengguna yang sudah melewati tahapan-tahapan registrasi.

### 3.2.3 Spesifikasi Proses (*Process Specification*)

#### 3.2.3.1 Spesifikasi Proses (*Process Specification*) DFD Level 0

Berdasarkan pada tabel 3.6, akan menjelaskan spesifikasi dari table DFD level 0

Table 3.6: *tabel spesifikasi DFD level 0*

No uji	Proses	Masukan	Keluaran	Logika Proses
1	registrasi2	pengisian data	konf data	Begin IF login dengan gmail THEN Mendapatkan akun ELSE isi data diri Endif End
2	verifikasi email	verifikasi kode	konf kode	Begin IF Pilih menu: email terverifikasi THEN Menampilkan aktifasi akun Endif End
3	aktifasi akun	konf akun	aktifasi akun	Begin IF akun diaktivasi THEN Akun Aktif Endif End

### 3.2.4 Kamus Alir Data(*Data Flow Dictionary*)

Pada tabel 3.7 akan menjelaskan tentang kamus aliran data pada proses registrasi.

Table 3.7: Kamus Aliran Data

No uji	Nama Alir Data	Keterangan
1	Pengisian Data	/*Pengisian Data <i>User</i> Dengan Gmail*/
2	Konfirmasi Akun	/*Konfirmasi Akun <i>User</i> */
3	Konfirmasi Data	/*Konfirmasi Data <i>User</i> */
4	Aktifasi Akun	/*Pengaktifan Akun <i>User</i> */

## 3.2.5 Perancangan Database

### 3.2.5.1 Conceptual Data Model

Pada gambar 3.5 adalah perancangan *database* registrasi pada aplikasi CGV-blitz dalam benduk *Conceptual Data Model* (CDM).

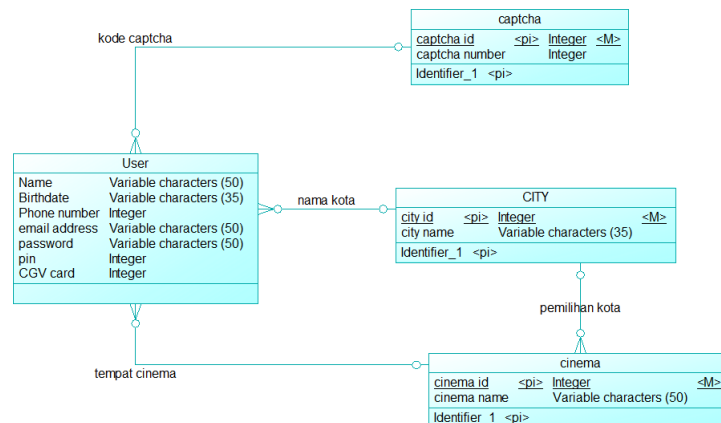


Figure 3.5: CDM

### 3.2.5.2 Physical Data Model

Pada gambar 3.6 merupakan perancangan *database* registrasi pada aplikasi CGV-blitz dalam benduk *Physical Data Model* (PDM)

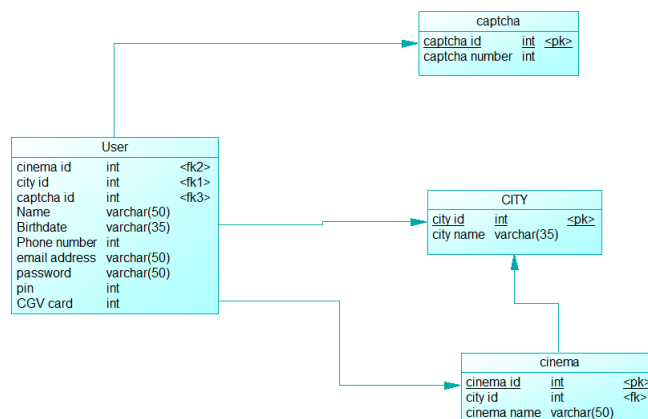


Figure 3.6: PDM

### 3.2.5.3 Kamus Data Tabel(Database)

Pada tabel 3.8 ; 3.9 ; 3.10 ; 3.11, menjelaskan tentang kamus data tabel *database*.

Table 3.8: *User*

No	Field	Type	Panjang	keterangan
1	<i>Name</i>	<i>Variable Character</i>	50	-
2	<i>Birthdate</i>	<i>VarChar</i>	35	tanggal lahir
3	<i>Phone Number</i>	<i>Integer</i>		nomor telpon
4	<i>email address</i>	<i>VarChar</i>	50	<i>email user</i>
5	<i>Password</i>	<i>VarChar</i>	50	
6	<i>Pin</i>	<i>integer</i>		
7	<i>CGV card</i>	<i>integer</i>		

Table 3.9: *Captcha*

No	Field	Type	Panjang	keterangan
1	<i>Captcha ID</i>	<i>Integer</i>		
2	<i>Captcha number</i>	<i>Integer</i>		Kode <i>Captcha</i>

Table 3.10: *City*

No	Field	Type	Panjang	Keterangan
1	<i>City ID</i>	<i>Integer</i>		
2	<i>City name</i>	<i>Integer</i>		Nama kota

Table 3.11: *Cinema*

No	Field	Type	Panjang	keterangan
1	<i>Cinema ID</i>	<i>Integer</i>		
2	<i>Cinema name</i>	<i>VarChar</i>	35	Nama Bioskop

### 3.2.6 Struktur Menu

Berikut pada gambar 3.7, menggambarkan struktur menu yang ada di aplikasi CGV-blitz.

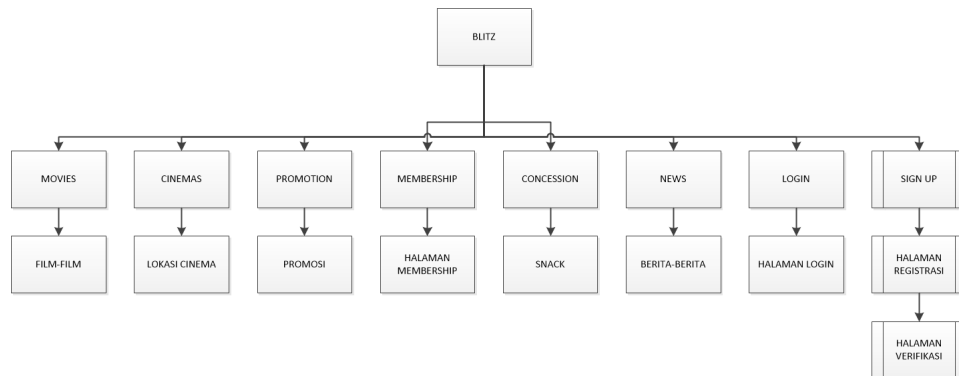


Figure 3.7: *struktur menu*

Berdasarkan pada gambar 3.7, laporan ini hanya akan membahas bagaimana proses registrasi blitz berjalan. Didalam bagan ini proses registrasi ditandai dengan model subproses, yang bentuknya seperti pada gambar 3.8 berikut yang menggambarkan bagaimana bentuk subproses itu sendiri.



Figure 3.8: *sub proses*

### 3.2.7 Perancangan Antar Muka

Berikut adalah rincian *coding* perancangan antar muka yang akan dibangun.

#### 3.2.7.1 Index

Berdasarkan pada gambar 3.9 , merupakan fungsi yang nantinya akan menampilkan tampilan registrasi menggunakan Google+. dengan rincian sebagai berikut.

Berdasarkan pada gambar 3.10 , fungsi *PHP* disini digunakan untuk menghubungkan fungsi *coding* pada halaman lain, dan ditempatkan sebelum *coding HTML*. Fungsi dari *coding* tersebut dapat dijabarkan sebagai berikut.

```

1  <?php
2      require_once "config.php";
3      if (isset($_SESSION['access_token'])) {
4          header('location: login.php');
5          exit();
6      }
7      $loginURL = $client->createAuthUrl()
8  ?>
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <title>Login</title>
14 </head>
15 <body>
16 <input type="button" onclick="window.location = '<?php echo $loginURL ?>';" value="login with google">
17 </body>
18 </html>
19

```

Figure 3.9: Fungsi Index

```

1  <?php
2      require_once "config.php";
3      if (isset($_SESSION['access_token'])) {
4          header('location: login.php');
5          exit();
6      }
7      $loginURL = $client->createAuthUrl()
8  ?>

```

Figure 3.10: Fungsi PHP Pada Index

1. "require once" : Membutuhkan suatu file agar fungsi tersebut bisa berjalan, apabila file tersebut tidak ditemukan maka fungsi selanjutnya tidak akan dijalankan.
2. "if" : Fungsi untuk permissalan
3. "isset" : Untuk meng-"set" suatu variable yang nantinya akan dijalankan, jika variable berhasil di set maka akan mengembalikan nilai *TRUE*, jika sebaliknya maka akan mengembalikan nilai *FALSE*.
4. "session" : Untuk mengingat data yang sudah diinputkan oleh *user*.
5. "header" : Untuk mengarah ke fungsi *coding* yang lain.
6. "exit" : Untuk keluar dari fungsi tersebut.
7. "*LoginURL = client-createAuthUrl()*" : Variabel dari *LoginURL* memiliki nilai berupa *variable client* yang mengarah ke *method createAuthUrl()*.

Setelah memasukkan fungsi *PHP* selanjutnya diikuti oleh *coding HTML* seperti pada gambar 3.11 berikut.

*Coding HTML* disini berfungsi untuk membuat tampilan *user interface*. berikut penjabaran *fungsi coding HTML*.

```

10 <!DOCTYPE html>
11 <html>
12 <head>
13     <title>Login</title>
14 </head>
15 <body>
16     <input type="button" onclick="window.location = '<?php echo $loginURL ?>';" value="login with google">
17 </body>
18 </html>

```

Figure 3.11: *HTML Pada Index*

1. "html" : Untuk memulai fungsi *HTML*
2. "head" : Header pada *HTML*
3. "title" : Untuk menempatkan judul pada *coding HTML*
4. "body" : Untuk "isi" yang ada pada *HTML*
5. "input type....login with google" : Untuk membuat tombol login yang nantinya jika ditekan akan mengarah ke tampilan *log in google*

### 3.2.7.2 Config

```

<?php
    session_start();
    require_once('GoogleAPI/vendor/autoload.php');
    $client = new Google_Client();
    $client->setClientId("628097087661-46ku15rmmfkb3mevobksnigdcqh11sq.apps.googleusercontent.com");
    $client->setClientSecret("wNcWzMB3AllNviRWftJWzoen");

    $client->setRedirectUri("http://localhost/registrasi/callback.php");
    $client->setAuthConfig('client_credentials.json');
    $client->addScope("https://www.googleapis.com/auth/plus.login https://www.googleapis.com/auth/userinfo.email")

?>

```

Figure 3.12: *Fungsi PHP Pada Config*

Berdasarkan pada gambar 3.12, fungsi config ini memegang peran penting dalam berjalannya rencangan aplikasi ini, karena file config ini mengandung fungsi yang menghubungkan aplikasi kepada Google+ *API*. Seperti *coding* yang tertera berikut

*GoogleAPI/vendor/autoload.php*

Pada *coding* tersebut kita memanggil semua fungsi yang dibutuhkan untuk menyambungkan aplikasi kepada google+ *API*.

### 3.2.7.3 Callback

Berdasarkan pada gambar 3.13, fungsi file *callback* disini adalah untuk melakukan pengecekan apakah akun *user* sudah pernah diregistrasikan atau belum, berikut

```

1  <?php
2  require_once('config.php');
3  if (isset($_SESSION['accessToken']))
4      $client->setAccessToken($_SESSION['accessToken']);
5  else if (isset($_GET['code'])) {
6      $token = $client->fetchAccessTokenWithAuthCode($_GET['code']);
7      $_SESSION['accessToken'] = $token;
8  }else {
9      header("location: index.php");
10     exit();
11 }
12
13 $oAuth = new Google_Service_Oauth2($client);
14 $user = $oAuth->userinfo_v2_me->get();
15
16 $_SESSION['uid'] = $userData['uid'];
17 $_SESSION['email'] = $userData['email'];
18 $_SESSION['gender'] = $userData['gender'];
19
20
21 header("location: login.php");
22 exit();
23 ?>

```

Figure 3.13: *Fungsi PHP Pada Callback*

penjelasan logika dari file *callback* tersebut

1. "if" (jika) akun tersebut sebelumnya sudah pernah diregistrasi, maka akan langsung dialihkan ke bagian *login.php*
2. "if else" (jika tidak) akun tersebut akan didaftarkan, jika berhasil didaftarkan maka akan langsung menuju ke bagian *login.php*
3. "else" (tidak) jika proses registrasi gagal maka akan dialihkan ke bagian *index.php*

Dalam fungsi ini juga digunakan *methodget()*, *method* ini digunakan untuk mengambil info data pada akun *user*, dalam fungsi coding pada gambar 3.13 data yang diambil berupa *uid*, *email*, dan juga *gender*. Dan disimpan menggunakan fungsi *session*.

#### 3.2.7.4 Login

Berdasarkan pada gambar 3.14, fungsi ini akan menampilkan informasi yang sebelumnya telah disimpan dan dijelaskan pada gambar 3.13. terdapat 2 fungsi pada coding-an ini, pertama adalah *coding fungsi PHP* yang diletakkan sebelum fungsi *HTML*, dan *coding HTML* yang digunakan sebagai fungsi untuk *user interface*. Berikut adalah penjabaran *fungsi-fungsi* tersebut.



```

1  <?php
2      session_start();
3
4      if (!isset($_SESSION['access_token'])) {
5          header('location: login.php');
6          exit();
7      }
8  ?>
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <title></title>
14 </head>
15 <body>
16
17 <tr>
18     <td>UID</td>
19     <td><?php echo $_SESSION['uid'] ?></td>
20 </tr>
21
22
23 <tr>
24     <td>email</td>
25     <td><?php echo $_SESSION['email'] ?></td>
26 </tr>
27
28 <tr>
29     <td>gender</td>
30     <td><?php echo $_SESSION['gender'] ?></td>
31 </tr>
32
33 </body>
34 </html>

```

Figure 3.14: *Fungsi Pada Login*

```

1  <?php
2      session_start();
3
4      if (!isset($_SESSION['access_token'])) {
5          header('location: login.php');
6          exit();
7      }
8  ?>
9

```

Figure 3.15: *Fungsi PHP Pada Login*

1. Pada gambar 3.15, fungsi yang digunakan adalah untuk melakukan pengecekan terhadap *session* yang sudah di set, jika tidak ditemukan maka *user* akan dialihkan kembali ke *login.php* .

```
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13     <title></title>
14 </head>
15 <body>
16
17 <tr>
18     <td>UID</td>
19     <td><?php echo $_SESSION['uid'] ?></td>
20 </tr>
21
22
23 <tr>
24     <td>email</td>
25     <td><?php echo $_SESSION['email'] ?></td>
26 </tr>
27
28 <tr>
29     <td>gender</td>
30     <td><?php echo $_SESSION['gender'] ?></td>
31 </tr>
32
33 </body>
34 </html>
```

Figure 3.16: Fungsi HTML Pada Login

2. Fungsi *HTML* yang digunakan disini untuk menampilkan hasil tampilan data yang sudah disimpan berdasarkan gambar 3.13. Penjabaran fungsi tambahan yang ada pada gambar 3.16 adalah sebagai berikut.

- (a) "*tr*" : Untuk membuat baris tabel.
- (b) "*td*" : Untuk membuat data tabel.
- (c) "*echo session (gender/email/uid)*" : Untuk memanggil data yang disimpan di *session*.

### 3.2.7.5 Dbconfig

```
<?php
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root'); // DB username
define('DB_PASSWORD', ''); // DB password
define('DB_DATABASE', 'google'); // DB name
$connection = mysql_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD) or die( "Unable to connect");
$dbdatabase = mysql_select_db(DB_DATABASE) or die( "Unable to select database");
?>
```

Figure 3.17: Fungsi PHP Pada Dbconfig

Berdasarkan pada gambar 3.17, fungsi dari codingan *PHP* di file *Dbconfig* ini adalah untuk mengkoneksikan aplikasi rancangan kita ke database. Perincian fungsi *php* tersebut adalah sebagai berikut

1. "*define('DB SERVER, localhost').....('DB DATABASE','google')*" : Untuk membedakan *database localhost*, beserta *input username,password*, dan nama tabel
2. "*mysql connect*" : Untuk menenghubungkan *database* ke *localhost*, beserta *input username*, dan *password*. Dan jika gagal maka error
3. "*mysql select db*" : Untuk Menentukan tabel mana yang akan digunakan, jika gagal maka error.

### 3.2.7.6 Function

```

1  <?php
2  require 'dbconfig.php';
3  function checkuser($uid,$email,$gender){
4      $check = mysql_query("select * from google where uid='$uid'");
5      $check = mysql_num_rows($check);
6      if (empty($check)) { // if new user . Insert a new record
7          $query = "INSERT INTO google (uid,email,gender) VALUES ('$uid','$email','$gender')";
8          mysql_query($query);
9      } else { // If Returned user . update the user record
10         $query = "UPDATE google SET email='$email' , gender='$gender' where uid='$uid' ";
11         mysql_query($query);
12     }
13 }

```

Figure 3.18: Fungsi *PHP* Pada Function

Mengacu pada gambar 3.18, fungsi *file function* disini adalah sebagai sambungan dari pengkoneksian *database*, dengan perincian fungsi sebagai berikut

#### Fungsi SQL

1. *Select* : Memilih nama tabel yang akan digunakan.
2. *Insert* : Memasukkan nilai kedalam tabel yang akan dipilih.
3. *Update* : Memperbarui data terbaru yang diinput.

#### Fungsi Menyeluruh

1. *if* (jika) data yang diinputkan sebelumnya tidak ada/kosong, maka akan menginputkan data baru ke dalam tabel *google*
2. *else* (lainnya) mengupdate data lama yang sebelumnya disimpan didalam *database*.

### 3.2.7.7 Pembuatan Database

```
1 CREATE TABLE google(  
2 'id' INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
3  
4 'UID' VARCHAR(30) NOT NULL,  
5  
6 'email' VARCHAR(50) NOT NULL,  
7  
8 'gender' VARCHAR(20) NOT NULL,  
9 )
```

Figure 3.19: Pembuatan Database

Berdasarkan pada gambar 3.19, *database* dapat dijabarkan pengertiannya sebagai berikut

1. "id" : Memiliki tipe data **INTEGER** dan bertipe **NOT NULL** serta akan menambah seiring adanya data baru (*AUTO INCREMENT*), dan berstatus **PRIMARY KEY** yang memiliki arti bahwa id ini merupakan pembeda dari atribut pada tabel-tabel yang lain
2. "UID" : Memiliki tipe data **VARCHAR** yang memiliki panjang sepanjang 30 dan bertipe **NOT NULL**
3. "email" : Memiliki tipe data **VARCHAR** yang memiliki panjang sepanjang 50 dan bertipe **NOT NULL**
4. "gender" : Memiliki tipe data **VARCHAR** yang memiliki panjang sepanjang 20 dan bertipe **NOT NULL**

Berdasarkan data diatas, kata yang bercetak tebal memiliki arti sebagai berikut.

1. **INTEGER** : Data yang dimasukkan hanya berupa angka tidak bisa yang lain.
2. **VARCHAR** (*Variable Character*) : Data yang dimasukkan berupa *character*.
3. **NOT NULL** : Data yang akan diisi tidak boleh kosong.

# BAB IV

## IMPLEMENTASI DAN PENGUJIAN

### 4.1 Lingkungan Implementasi

Sesudah menyelesaikan proses analisis, proses yang dilakukan selanjutnya adalah perancangan spesifikasi. Perancangan spesifikasi ini dilakukan untuk merancang kebutuhan-kebutuhan perangkat keras dan perangkat lunak pada aplikasi yang dibangun.

#### 4.1.1 Perangkat Keras

Berikut pada tabel 4.1 merupakan spesifikasi yang digunakan pada proses penelitian proyek I.

Table 4.1: Deskripsi Perangkat Keras

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Harddisk</i>	500 GB	Media untuk menyimpan data aplikasi yang dibuat
2	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan
3	<i>processor</i>	Intel® core™ i5-8250U CPU @ 1.60GHz (8 CPUs), 1.8Ghz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4	Infrastruktur Jaringan		Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses

#### 4.1.2 Perangkat Lunak

Berikut pada tabel 4.2 merupakan spesifikasi yang digunakan pada proses penelitian proyek I.

### 4.2 Pembahasan Hasil Implementasi

Berdasarkan pada gambar 3.1, dapat dipahami bahwa proses registrasi pada aplikasi CGV-blitz masih memiliki ruang untuk dikembangkan, diantaranya melakukan registrasi menggunakan akun Google+, registrasi akan jauh lebih efektif dan efisien.

Table 4.2: Deskripsi perangkat Lunak *Client*

No	Tools	Fungsi	Keterangan
1	<i>Windows 10</i>	Sistem Operasi	-
2	<i>Xampp 1.7.3</i>	<i>server</i> Basis Data	-
3	<i>Photoshop CS</i>	desain antar muka	-
4	<i>PHP,HTML,CSS</i>	<i>Bahasa Program</i>	-
5	<i>sublime text</i>	software pendukung	-
6	<i>google chrome</i>	<i>browser</i>	-

Berdasarkan pada gambar 3.19 , fungsi-fungsi bahasa SQL yang digunakan adalah **INSERT**, dan **UPDATE** . Dan bahasa pemrograman *PHP* yang digunakan untuk menghubungkan *website* dengan database.

Berikut berdasarkan pada gambar 4.1 dan 4.2 merupakan hasil dari implementasi sistem yang dirancang.

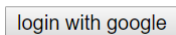


Figure 4.1: *Halaman Login*

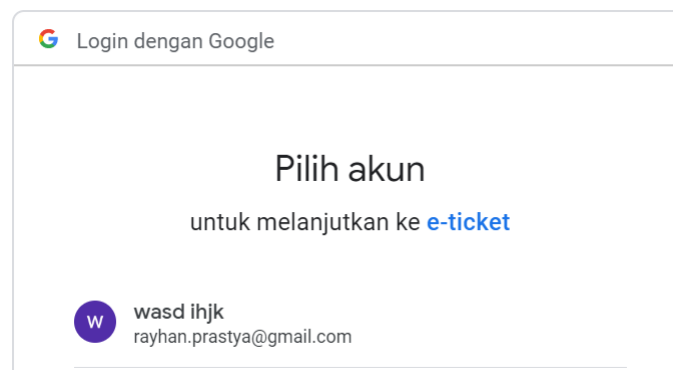


Figure 4.2: *Masuk Akun Google*

## 4.3 Pengujian

### 4.3.1 *Black Box*

Berikut pada tabel 4.3, merupakan pengujian dengan melakukan metode *blackbox*

Table 4.3: Blackbox

Kelas	Butir Uji	Identifikasi	Tingkat Pengujian	Jenis Pengajuan	Jadwal
		PDHUPL			
Lingkungan	Halaman Pendaftaran	A_01	Pengujian Sistem	Black Box	19/07/2019

Table 4.4: *Masuk Akun Google*

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria evaluasi	Hasil yang didapat	Kesimpulan
A_01	Halaman	Calon User	Calon User menginputkan data diri (Preferred Cinema All=Bandung, Select Cinema= Paris Van Java, Full Name= Rayhan Prastya, Gender= Select Gender Laki-laki, Birthdate=30 January 2001, Phone Number= 08215205****, City= Bandung, Email Address= Jonijoniwak****@gmail.com, New Password= rayhana******, New Pin= 331100, Captcha= 8)	Data Tersimpan di Database	-	Calon User mendapatkan akun	Diterima

# BAB V

## KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari Laporan project Analisis Sistem Registrasi Pada Aplikasi *Online Ticketing* CGV Blitz dapat diambil kesimpulan bahwa:

1. Proses Registrasi Pada Aplikasi CGV-Blitz sudah cukup baik, namun masih ada hal yang dapat dikembangkan.
2. Bahasa pemrograman *PHP* digunakan untuk menyambungkan Fungsi-Fungsi dari file yang lain, dan *MySQL* digunakan sebagai tempat penyimpanan data pengguna.

### 5.2 Saran

Agar kerja dari Sistem Registrasi Pada Aplikasi *Online Ticketing* CGV-Blitz yang dirancang lebih optimal, maka penulis memberikan beberapa saran yaitu:

1. Untuk lebih menyempurnakan Sistem Registrasi Pada Aplikasi *Online Ticketing* yang telah dibangun, diharapkan kedepannya program aplikasi ini bisa lebih di kembangkan kembali.



## DAFTAR PUSTAKA

- [1] Yuni Astutik and Siti Nuriyatin. Analisis kesalahan siswa dalam menyelesaikan soal cerita aritmatika sosial. *Jurnal Pendidikan Matematika*, 4(2), 2019.
- [2] Esmeralda Crespo-Almendros and S Del Barrio-García. Online airline ticket purchasing: Influence of online sales promotion type and internet experience. *Journal of Air Transport Management*, 53:23–34, 2016.
- [3] Jeperson Hutahaeen and Evi Ariyanti Purba. Rancangan bangun e-ticket bioskop dengan metode waterfall berbasis web. *Riau Journal Of Computer Science*, 2(2):51–58, 2016.
- [4] Irma Devi Lestari. Klasifikasi online dan google. 2016.
- [5] Rivaldi Masuara, Rizal Sengkey, and Virginia Tulenan. Rancang bangun e-ticketing bioskop studio 21 manado berbasis multiplatform. *Jurnal Teknik Elektro dan Komputer*, 4(2):41–55, 2015.
- [6] Rosdania Rosdania, Fahrul Agus, and Awang Harsa Kridalaksana. Sistem informasi geografi batas wilayah kampus universitas mulawarman menggunakan google maps api. *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 10(1):38–46, 2016.
- [7] Nur Rubiati. Aplikasi informasi pelayanan fitness pada golden fitness center dumai dengan bahasa pemrograman php. *INFORMATIKA*, 10(1):1–6, 2018.
- [8] Rina Candra Noor Santi and Sri Eniyati. Implementasi statistik dengan database mysql. *Dinamik*, 20(2), 2015.
- [9] Vicki L Sauter. Making data flow diagrams accessible for visually impaired students using excel tables. *Journal of Information Systems Education*, 26(1), 2015.

- [10] Imam Solikin. Implementasi e-modul pada program studi manajemen informatika universitas bina darma berbasis web mobile. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2(2):492–497, 2018.
- [11] Rini Sovia and Jimmy Febio. Membangun aplikasi e-library menggunakan html, php script, dan mysql database. *Jurnal Processor*, 6(2), 2017.
- [12] Yunita Trimarsiah and Muhajir Arafat. Analisis dan perancangan website sebagai sarana informasi pada lembaga bahasa kewirausahaan dan komputer akmi baturaja. *JURNAL MATRIK*, 19(1):01–10, 2017.
- [13] Hao Xiong, Haocheng Zhang, Xiaoju Dong, Lingxi Meng, and Wenyang Zhao. Dfdvis: A visual analytics system for understanding the semantics of data flow diagram. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pages 660–673. Springer, 2017.