BAB II
DASAR TEORI

BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Tujuan penelitian ini adalah untuk memfasilitasi proses pendaftaran peserta tes secara daring, sehingga dapat membantu proses agar lebih mudah dan cepat serta dapat diakses kapan saja dan di mana saja. Untuk mendukung pengembangan Sistem Informasi Pendaftaran EPT-P, terdapat beberapa penelitian yang dapat dijadikan sebagai sumber referensi. Penelitian yang berjudul Rancang Bangun Sistem Informasi TOEFL pada Pusat Bahasa Universitas Mataram (Imam Tantowi, Moh.Ali Albar, & Fitri Bimantoro, 2019) [3]. Di mana pada Pusat Bahasa Universitas Mataram masih berjalan secara manual. Mulai dari pendaftaran, penjadwalan, pembagian ruang ujian dan pengumuman hasil tes TOEFL yang hanya bisa dilihat di pusat Bahasa Universitas Mataram yang tentunya membutuhkan banyak waktu bagi peserta maupun petugas pusat bahasa. Sehingga, dibuat rancang bangun sistem informasi TOEFL untuk memudahkan pelayanan pendaftaran, penjadwalan, pembagian ruang ujian dan pengumuman hasil tes TOEFL pada Pusat Bahasa Universitas Mataram. Sistem ini menggunakan PHP dan HTML sebagai bahasa pemrograman, dengan *framework codeigniter*, dan MySQL sebagai *server database*. Sedangkan untuk pengembangan nya, sistem ini menggunakan metode *waterfall* dan *blackbox testing*.

Adapun penelitian lain dengan judul Sistem Informasi TUTEP dan TOEFL Online pada UPT Bahasa Tanjungpura menggunakan metode *user centered design* (Edwin Rianto, Ilhamsyah, & Dwi Marisa Midyanti, 2021) [4]. Pada UPT Bahasa Tanjungpura, proses pelayanan pelatihan dan tes bahasa asing masih bersifat manual. Dari pendaftaran, penjadwalan, pengumuman, dan sertifikasi masih menggunakan *paper-based*. Sehingga, tidak efektif bagi pendaftaran karena harus ke UPT Bahasa terlebih dahulu. Oleh karena itu UPT Bahasa Universitas Tanjung pura merancang dan membangun sistem informasi pendaftaran TUTEP dan TOEFL ITP menggunakan metode *user centered design*. Dengan *framework laravel* dan pengujian fungsionalitas.

Penelitian selanjutnya dengan judul Sistem Informasi Pelayanan dan Pelaksanaan Test TOEFL berbasis *Website* pada program studi Sistem Informasi Universitas Komputer Indonesia (Sekar Rhiandari

Graitasadu, 2017) [5]. Pada program studi sistem informasi di Universitas Komputer Indonesia, salah satu syarat untuk menyusun skripsi adalah minimal skor TOEFL > 425. Pada studi kasus ini, pendaftar harus datang ke bagian sekretariat jurusan dengan membawa formulir dan pas foto. Petugas pendaftaran juga masih mengolah data pendaftaran, mahasiswa, jadwal tes, dan pembayaran secara semi-manual. Pengumuman hasil tes yang hanya dapat dilihat di papan pengumuman kampus. Proses validasi nilai, yang harus menemui koordinator program studi Sistem Informasi. Hal – hal di atas mengakibatkan kurang efisiensi waktu dan memakan biaya relatif lebih banyak. Oleh karena itu, dibuat sistem informasi untuk proses pendaftaran dan pengolahan data dapat lebih baik. Sistem ini dibuat menggunakan sistem *object-oriented*, *framework codeignniter*, dan MySQL sebagai *server database*. Sedangkan untuk pengembangan nya, sistem ini menggunakan metode prototipe dan *blackbox testing*.

Pada penelitian ini yang berbeda dengan penelitian sebelumnya, yaitu Sistem Informasi Pendaftaran EPT-P tes berbasis *website*, Studi kasus UPT Bahasa Politeknik Negeri Cilacap yang dikembangakan menggunakan bahasa pemrograman PHP, *framework laravel*, dan database MySQL. Sistem informasi ini menggunakan metode *prototyping*, yang berfokus pada *end-user* agar dapat membahami sistem yang sedang dikembangkan. Sistem ini terdapat fitur remidial, di mana jika peserta harus mengikuti tes lagi, maka peserta hanya perlu melakukan pembayaran dan memilih jadwal tes. Sistem ini diharapkan dapat mempermudah pelayanan pendaftaran bagi peserta dan pengelolaan data bagi petugas UPT Bahasa Politeknik Negeri Cilacap. Berikut ini adalah tabel tentang penelitian yang terkait :

Tabel 2. 1 Tabel Perbandingan Penelitian

Komponen	Peneliti			
	Imam Tantowi, Moh.Ali Albar, & Fitri Bimantoro	Edwin Rianto, Ilhamsyah, & Dwi Marisa Midyanti	Sekar Rhiandari Graitasadu	
Tahun	2019	2021	2017	
Judul Penelitian	Rancangan Bangun Sistem Informasi TOEFL - Pusat Bahasa Univeritas Mataram	Sistem Informasi TUTEP & TOEFL Online - UPT Bahasa Tanjungpura	Sistem Informasi Pelayanan dan Pelaksanaan Tes TOEFL - Universitas Komputer Indonesia	Sistem Informasi Pendaftaran EPT-P Berbasis Website Pada UPT Bahasa Politeknik Negeri Cilacap
Bahasa Pemrograman	PHP	-	*object-oriented*	PHP
Framework	*CodeIgniter*	*Laravel*	*CodeIgniter*	*Laravel*
Basis Data	MySQL	-	MySQL	MySQL
Metode Pengembangan	*Waterfall*	*User Centered Design*	*Portotype*	*Portotype*
Metode Pengujian	*Blackbox*	Fungsionalitas	*Blackbox*	*Blackbox*

2.2. Landasan Teori

2.2.1 Sistem Informasi

Sistem merupakan beberapa elemen yang berhubungan dan berinteraksi untuk menjalankan proses dan tujuan dengan satu kesatuan[6]. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya Jogiyanto (2005). Sehingga sistem informasi adalah kegiatan yang mencakup dalam perolehan, melakukan kombinasi, penyimpanan dan penggunaan informasi untuk mencapai tujuan tertentu dibuatnya sebuah sistem[7]. Sistem informasi memeliki beberapa komponen[6], seperti:

a. Perangkat keras *(hardware*): mencakup alat - alat fisik seperti *computer* dan *printer*
b. Perangkat lunak (*software*) atau program: sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
c. Prosedur (*procedure*): sekumpulan turan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
d. Orang (*brainware*): semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
e. Basis data (*database)*: sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
f. Jaringan komputer dan komunikasi data: sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau di akses oleh sejumlah pemakai.

Adapun jenis – jenis sistem informasi dengan tujuan berbeda sesuai kebutuhan[8], sebagai berikut.

a. *Transaction Processing System* (TPS).

TPS adalah sistem informasi yang terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi bisnis rutin seperti daftar gaji dan inventaris.

b. *Office Automation System (OAS)* dan *Knowledge Work System (KWS)*.

OAS mendukung pekerja data, yang biasanya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau

memanipulasikannya dengan caracara tertentu sebelum menyebarkannya secara keseluruhan dengan organisasi Sedangkan *KWS* mendukung para pekerja professional seperti ilmuan, insinyur dan doktor dengan membantu menciptakan pengetahuan baru dam memungkinkan mereka memberikan kontribusi ke organisasi atau masyarakat

c. Sistem Informasi Manajemen.

Sistem ini menghasilkan informasi yang digunakan untuk membuat keputusan, dan juga dapat membantu beberapa fungsi informasi bisnis yang sudah terkomputerisasi.

d. *Decision Support System*.

Sistem ini menggunakan basis data sebagai sumber data yang menekan kan pada fungsi mendukung pembuat keputusan diseluruh tahapanya.

e. *Expert System (ES)* dan *Artificial Intelegent (AI)*.

AI digunakan untuk mengembangkan mesin secara cerdas. Cara melakukan risetnya adalah dengan memahami bahasa ilmiahnya dan menganalisis kemampuan berpikir melalui *problem* sampai kesimpulan logika nya.

f. Group Decision Support System (GDSS) dan Computer-Support Collaboration Work System (CSCW).

Sistem ini memuat solusi yang mencakup perangkat lunak yang disebut *"groupware"* untuk kolaborasi tim melalui komputer yang terhubung dengan jaringan.

g. Executive Support System.

Sistem ini berfungsi untuk mengatur interaksinya dengan lingkungan eksternal dengan menyediakan grafik – grafik dan pendukung komunikasi di tempat yang dapat di akses.

2.2.2 Rekayasa Perangkat Lunak

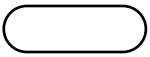
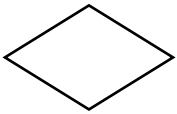
Perangkat lunak merupakan semua dokumentasi dan data yang berhubungan, bukan hanya sebuah program komputer. Dokumentasi dapat berupa dokumen desain, pengoperasian, dan pemeliharaan. Rekayasa perangkat lunak menurut (Sommerville, 2016), merupakan disiplin ilmu yang menjelaskan seluruh aspek produksi perangkat lunak, dari tahap analisis kebutuhan sistem hingga pemeliharaan sistem setelah digunakan[9].

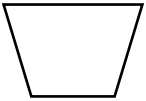
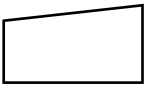
Representasi grafis, model dapat digunakan sebagai media untuk mendokumentasikan kebutuhan sistem agar dapat mudah dimengerti.

A. *Flowchart*

Flowchart adalah bagan alir yang menggunakan simbol untuk menggambarkan proses secara detail. flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan. Aktifitas yang di gambarkan didefinisikan dengan jelas sehingga dapat dimengerti oleh pembaca nya[9]. Simbol flowchart dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Flowchart*

No.	Simbol	Nama	Arti
1.		*Terminator*	Awal atau akhir konsep (prosedur)
2.		*Process*	Proses operasional
3.		*Document*	Dokumen atau laporan berupa *print out*
4.		*Decision*	Keputusan untuk situasi – situasi yang berbeda dnegan keputusan yang digambarkan.

5.		*Data*	*Input* dan *Output*
6.		*On-Page Reference/Connector*	Penghubung alur dalam halaman yang sama
7.		*Off-Page Reference/ Off-Page Connector*	Penghubung alur dalam halaman yang sama
8.		*Manual Opertaion*	Langkah proses yang tidak otomatis.
9.		*Manual Input*	Memasukan informasi secara manual.
10.		*Flow*	Arah alur dalam konsep

B. *Prototyping*

Prototyping merupakan metode pengembangan perangkat lunak di mana model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem[10]. Versi awal sistem ini merupakan perantara antara pengguna dan pengembang agar dapat berinteraksi dalam proses pengembangan. Berikut adalah tahapan metode *prototyping*:

a. *Requirments*

Pada tahap ini pengembang mendefinisikan kebutuhan dan informasi serta kegunaan dalam perangkat lunak. Metode Analisis Kebutuhan ini dapat dilakukan dengan cara wawancara, observasi dan dokumentasi. Kemudian data tersebut dikaji oleh pengembang agar dapat digunakan dengan baik.

b. *Quick Design*

Tahap ini adalah pembuatan desain sederhana yang akan memberi gambaran singkat tentang sistem yang akan dibuat. Pengembang akan melakukan desain berdasarkan dikusi yang telah dilakukan bersama klien.

c. *Build Prototipe*

Setelah *quick design* disetujui maka pengembang melakukan tahap pembangunan *prototipe*. Pembangunan prototipe merupakan tahap yang dijadikan sebagai rujukan bagi *programmer* untuk pembuatan program.

d. *User Evaluation*

Pada tahap ini klien mengevaluasi sistem yang telah dibuat dalam bentuk prototipe. Kemudian memberikan komentar dan saran terhadap apa yang telah dibuat.

e. *Refining prototipe*

Tahap ini merupakan revisi jika klien mempunyai catatan untuk perbaikan di dalam sistem. Tahap ini berulang sampai klien setuju dengan sistem yang akan dikembangkan.

f. *Implement and Maintain*

Pada tahap terakhir ini, produk akan segera dibuat oleh para programmer berdasarkan prototipe akhir. Kemudian sistem diuji dan diserahkan kepada klien.

C. *Blackbox*

Black-box testing merupakan metode pengujian perangkat lunak yang mencakup sebuah area dalam perangkat lunak (*Data Driven Testing*) hanya menguji kemampuan perangkat lunak menerima dan mengolah data masukan[11]. Metode ini tidak mengharuskan penguji mengetahui tentang program dan menguji seluruh bagian kode, serta memfokuskan pada hasil atau *output*[12]. Terdapat 10 jenis pengujian *black-box:*

a. *Equivalence Partitioning.*

Membagi masukan menjadi kelas data yang dapat digunakan untuk mengenerasi kasus uji.

b. *Boundary Value Analysis / Limit Testing.*

Mengizinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari Equivalence Partitioning.

c. *Comparison Testing.*

Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.

d. *Sample Testing.*

Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.

e. *Robustness Testing.*

Data input dipilih di luar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid

f. *Behavior Testing.*

Hasil uji tidak dapat di evaluasi jika hanya melakukan pengujian sekali, tapi dapat di evaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.

g. *Performance Testing.*

Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi.

h. *Requirement Testing.*

Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.

i. *Endurance Testing.*

Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.

j. *Cause – Effect Relationship Testing*.

Membagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja

2.2.3 Pemrograman Berorientasi Objek

Pendekatan berorientasi objek merupakan cara pengembang fokus pada pembangunan sistem yang lebih efisien, yang dapat

membuat analis bekerja dengan data dan sistem sebagai objek secara bersamaan[9].

A. UML *(Unified Modeling Language)*

UML *(Unified Modeling Language)* merupakan pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung[9]. Beberapa pemodelan yang termasuk UML *adalah use case diagram*, *class diagram*, *activity diagram*, dan *Squence diagram.*

a) ***Usecase* Diagram**

Diagram yang merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat[13]. Diagram ini digunakan untuk mengetahui fungsi apa yang ada pada sistem dan siapa yang dapat menggunakan nya. Simbol *usecase* diagram dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol *Usecase* Diagram

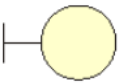
No.	Simbol	Nama	Arti
1.		Aktor	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*.
2.		*Use case*	Abstraksi dan interaksi antara sistem dan actor
3.		*Association*	Abstraksi dari penghubung antara actor dan *use case.*
4.		*Generalisasi*	Menunjukan spesialisasi actor untuk dapat berpartisipasi dengan *use case.*

5.	<<include>>	*Include*	Menunjukan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya
6.	<<extend>>	*Extend*	Menunjukan bahwa suatu *use case* merupakan tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi.

b) ***Sequence* Diagram**

Sequence diagram merupakan UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem, termasuk pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap waktu[14]. Simbol *sequence* diagram dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Simbol *Sequence* Diagram

No.	Simbol	Nama	Arti
1.		*Actor*	Menggambarkan orang yang berinteraksi dengan sistem
2.		*Entity Class*	Menggambarkan hubungan yang akan dilakukan
3.		*Boundary Class*	Menggambarkan sebuah gambaran dari form

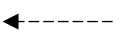
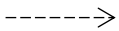
4.		*Control Class*	Menggambarkan hubungan antara *boundary* dengan tabel
5.		*Lifeline*	Menggambarkan tempat mulai dan berakhirnya *message*
6.		*Line Message*	Menggambarkan pengiriman *message* antar *class*

c) ***Class* Diagram**

Class diagram adalah diagram yang menggambarkan relasi antar kelas pada sistem serta bagaimana cara untuk saling berelasi[13]. Simbol *class* diagram dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol *Class* Diagram

No.	Simbol	Nama	Arti
1.		*Generalization*	Relasi antarkelas dengan makna generalisasispesialisasi (umum-khusus)
2.		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3.		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama 4. Realization

4.		*Realization*	Operasi yang dilakukan oleh suatu objek.
5.		*Dependency*	Relasi antarkelas dengan makna kebergantungan antarkelas.
6.		*Association*	Diagram yang menghubungkan antara objek satu dengan objek lainnya.

2.2.4 Basis Data

Basis data atau *database* merupakan kumpulan informasi yang disusun dan satu kesatuan yang utuh disimpan di dalam perangkat keras (komputer) secara sistematis sehingga dapat diolah oleh perangkat lunak dan menghasilkan informasi yang berguna[15]. Berdasarkan istilah data yang merupakan fakta yang dapat direkam dan disimpan pada media komputer. Data yang telah diolah menjadi sesuatu yang bermakna disebut informasi[16].

Pada penerapanya agar dapat mengelola basis data secara efisien dan mengubah data menjadi informasi yang bermakna, kita memerlukan alat yang mampu melakukan penangkapan dan pengambilan data yang dapat disebut dengan SQL (*Sturctured Query Language*). SQL adalah bahasa khusus yang digunakan untuk mengelola, mengambil, memanipulasi, dan menyimpan data dalam basis data[15]. Berikut merupakan pengelompokkan SQL:

a. *Data Definition Language* (DDL)

Data definition language merupakan perintah yang memiliki fungsi untuk mendefinisikan atribut basis data, seperti tabel dan kolom[15]. Perintah yang dapat digunakan adalah CREATE, ALTER, DROP.

b. *Data Manipulation Language (DML)*

Data manipulation language merupakan perintah pada umumnya yang berfungsi untuk pengambilan, penyisipan, pengubahan dan penghapusan data[15]. Perintah yang dapat dilakukan adalah INSERT, SELECT, DELETE, UPDATE.

c. *Data Access (Data Control Language / DCL)*

Data access merupakan perintah agar dapat melakukan pengendalian dan pengaksesan data[15]. Perintah yang dapat digunakan adalah GRANT (memberi kenadali pengkasesan data) dan REVOKE (mencabut kemampuan pengaksesan data).

d. *Data Integrity*

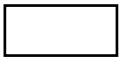
Data integrity merupakan perintah yang dapat digunakan untuk mengembalikan data sebelum terjadi kerusakan[15]. Perintah yang dapat digunakan adalah RECOVER TABLE.

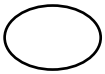
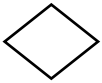
e. *Data Auxiliary*

Data auxiliary merupakan perintah untuk mengubah data atau kolom pada sebuah tabel[15]. Perintah yang dapat digunakan adalah SELECT INTO OUTFILE, LOAD, RENAME TABLE.

Dalam konteks implementasi basis data, konsep *Entity-Relationship Diagram* (ERD) menjadi penting. ERD digunakan untuk menggambarkan hubungan antar entitas serta bentuk fisik data yang akan disimpan dalam basis data. Sehingga model yang digunakan untuk menggambarkan bentuk fisik yang akan disimpan dalam *database* disebut *Entity Relation Diagram*[16]. Di mana terdapat entitas, atribut, dan hubungan antar entitas yang digambarkan. Simbol Entity Relation Diagram diagram dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Simbol *Entity Relation Diagram*

No.	Simbol	Nama	Arti
1.		Entittas	Entitas merupakan sesuatu yang nyata ataupun abstrak, yang dapat beruppa tempat, orang, kejadian, objek ataupun konsep.

2.		Atribut	Protperti atau karakteristik yang terdapat pada setiap entitas.
3.		Hubungan	Keterkaitan antara beberapa entitas.

2.2.5 EPT-P

Berasal EPT *(English Proficiency Test)* adalah tes menukur kemampuan bahasa inggris untuk kepentingan akademis. Keterampilan bahasa inggris yang diujikan adalah *reading*, *listening*, *speaking*, dan *writing*. Sedangkan materi yang diujikan adalah *grammar*, *vocabulary*, *pronuncation*, *intonation*, dan tekanan[1]. Hasil dari tes EPT merupakan perdiksi untuk nilai TOEFL yang menggunakan jenis soal *paper-based test*. Berikut merupakan konversi skor TOEFL[17], yang dapat dilihat pada Tabel 2.7.

Tabel 2. 7 Konversi Skor TOEFL

Jumlah Jawaban Benar	**Section I Listening**	**Section II Structure**	**Section III Reading**
50	68		67
49	66		66
48	64		65
47	63		63
46	62		61
45	61		60
44	60		59
43	59		58
42	58		57
41	57		56
40	56	67	55
39	56	66	54
38	55	64	53
37	55	63	52

36	54	61	51
35	52	59	50
34	52	58	49
33	51	57	48
32	50	55	48
31	50	54	48
30	49	53	47
29	49	52	47
28	48	51	46
27	48	50	45
26	47	49	45
25	46	48	44
24	46	47	43
23	45	46	42
22	44	45	41
21	44	44	41
20	43	43	40
19	43	42	39
18	42	41	38
17	41	41	37
16	41	41	36
15	40	40	35
14	39	39	34
13	38	38	33
12	37	37	32
11	36	36	31
10	34	34	30
9	33	33	29
8	32	32	28
7	31	30	27
6	30	28	26
5	29	26	25
4	28	25	24
3	27	24	23
2	26	22	22
1	25	20	28

Cara menghitunnya dengan menjumlahkan hasil konversi section I, section II, dan section III[17], seperti Tabel 2.8 berikut:

Tabel 2. 8 Contoh Hasil Perhitungan Skor TOEFL

Tipe Soal	Jumlah	Hasil Konversi
Section I Listening	30	49
Section II Structure	28	51
Section III Reading	43	48
Jumlah Skor Konverensi		148

Dari hasil jumlah koversi skor dapat diketahui skor tes dengan dikali 10, kemudian dibagi 3, maka diperoleh skor tes 493. Sama halnya pada UPT Bahasa Politeknik Negeri Cilacap terdapat EPT-P yang merupakan kepanjangan dari *English Proficiency Test* – Politeknik Negeri Cilacap. Di mana nilai tes EPT merupakan salah satu syarat program diploma dan sarjana terapan di Politeknik Negeri Cilacap. Namun, pada pelaksanaan dan pelayanan masih bersifat *semi-manual*. Seperti proses pendaftaran yang kurang efektif dan efisien baik dari segi ekonomi maupun waktu. Hal ini dikarenakan calon peserta harus datang ke UPT Bahasa untuk mengisi formulir.

Pada tahap pembayaran, calon peserta diwajibkan melampirkan bukti bayar untuk mendapatkan kartu tes, yang harus diambil di UPT Bahasa. Kemudian pada tahap penjadwalan, staf menjadwalkan tes serta menginformasikan jadwal setelah kuota terpenuhi. Oleh karena itu, dibuat sistem informasi pendaftaran tes EPT-P berbasis *website* untuk mempermudah bagi calon peserta maupun staf pengelola.

~Halaman Ini Sengaja Dikosongkan~