

# SISTEM INFORMASI PERHITUNGAN RENCANA ANGGARAN BIAYA PERENCANAAN BANGUNAN BERBASIS WEB MENGGUNAKAN METODE FRAMEWORK CODEIGNITER 3

Sawindri<sup>1)</sup>, Agung Triayudi<sup>2\*)</sup>

<sup>1, 2)</sup>Teknik Informatika, Universitas Nasional

Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta

e-mail: [sawindriindri@gmail.com](mailto:sawindriindri@gmail.com)<sup>1)</sup>, [agungtriayudi@civitas.unas.ac.id](mailto:agungtriayudi@civitas.unas.ac.id)<sup>2)</sup>

\*Penulis Korespondensi

## ABSTRAK

*Estimasi perencanaan perhitungan biaya bangunan yang akan dibutuhkan nantinya disebut dengan RAP (Rencana Anggaran Pelaksanaan). RAP merupakan tahapan perencanaan mengatur kebutuhan material dan tenaga kerja secara detail untuk tahapan awal sebelum melakukan pembangunan proyek. Sistem aplikasi ini dibangun dan didesain dengan menggunakan framework codeigniter 3. Untuk bahasa pemrograman yang digunakan adalah bahasa pemrograman php (Hypertext Preprocessor), dan untuk tempat penyimpanan database-nya menggunakan software XAMPP (phpMyAdmin). Framework codeigniter 3 merupakan sebuah kerangka dasar pembangun sistem informasi web, yang didalamnya terdapat kumpulan class dan function serta konsep yang telah ditata dengan struktur arsiteknya pada konsep MVC (model view controller). Aplikasi RAB ini bertujuan untuk mendata dan menghitung berapa jumlah anggaran biaya dan berapa lama proses berlangsungnya pekerjaan akan selesai sesuai target pelaksana. Adapun aplikasi perhitungan ini ditujukan untuk menghitung estimasi "Perencanaan Desain Berulang Pembangunan Rumah Susun Tahun 2019". Pengujian pada sistem aplikasi RAB dilakukan dengan menggunakan pengujian Black-Box, dimana dengan pengujian ini dapat dipastikan apakah sistem aplikasi web ini berjalan dengan baik ataupun mengalami kendala pada saat penggunaan menu dan tools yang ada pada sistem.*

**Kata Kunci:** Aplikasi Web, Framework Codeigniter, Konsep MVC.

## ABSTRACT

*Estimated planning for the calculation of building costs that will be needed later is called RAP (Implementation Budget Plan). RAP is the planning stage to regulate material and labor requirements in detail for the initial stages before carrying out project development. This application system is built and designed using the CodeIgniter 3 framework. The programming language used is the PHP (Hypertext Preprocessor) programming language, and the XAMPP (phpMyAdmin) software for database storage. The codeigniter 3 framework is a basic framework for building web information systems, in which there is a collection of classes and functions as well as concepts that have been arranged with the architectural structure on the MVC (model view controller) concept. This RAB application aims to record and calculate the amount of the budget and how long the process will take to complete the work according to the implementing target. The application for this calculation is intended to calculate the estimate of "Planning for Repeated Design of Flats for 2019". Testing on the RAB application system is carried out using Black-Box testing, where with this test it can be ascertained whether the web application system is running well or experiencing problems when using the menus and tools on the system.*

**Keywords:** Framework Codeigniter, MVC Concept, Web Application.

## I. PENDAHULUAN

Rencana anggaran biaya bangunan (RAB) adalah sebuah perhitungan banyaknya biaya yang akan dibutuhkan untuk keperluan pengeluaran biaya bahan, alat, dan upah bagi pekerja pelaksana pembangunan proyek. Menurut Bachtiar Ibrahim dalam buku Rencana dan *Estimate Real of Cost*, 1993, RAB adalah jumlah banyaknya biaya yang dibutuhkan untuk bahan serta upah, dan biaya lain yang berhubungan dengan pelaksanaan atau pekerjaan proyek bersangkutan [1]. Perencanaan RAB suatu bangunan dibutuhkan karena untuk mengantisipasi pengeluaran biaya. Kebutuhan biaya yang biasanya diperlukan adalah biaya bahan, alat dan upah untuk para pekerja. Perhitungan RAB biasanya menggunakan *ms. Excel* untuk menghitung keseluruhan biaya, seperti volume, harga bahan, alat, dan daftar harga upah yang akan digunakan dalam perhitungan total biaya keseluruhan. Menghitung RAB dapat dilakukan dengan menghitung luas bangunan atau volume bangunan kemudian luas/volume tersebut dapat dikalikan dengan harga satuan yang telah ditetapkan SNI (Standart Nasional Indonesia) dari harga upah dan harga bahan material yang digunakan atau dibutuhkan [2].

System informasi dengan seiring berkembangnya teknologi yang sangat pesat banyak digunakan untuk membantu mempermudah pekerjaan sehari-hari ataupun dalam dunia kerja. System informasi (aplikasi) berupa

situs media online yang digunakan untuk mencari informasi atau bahkan dapat digunakan untuk memecahkan permasalahan yang banyak orang sulit untuk menyelesaikannya. Pada *system* informasi (aplikasi) penelitian kali ini penulis bermaksud membangun *system* informasi (aplikasi) yang sistemnya dapat memasukkan data dan menghitung jumlah total harga dengan hasil yang lebih efektif. Penulis bermaksud membangun sebuah “*System Informasi Perhitungan Rencana Anggaran Biaya Perencanaan Bangunan Berbasis Web Menggunakan Metode Framework Codeigniter 3*”.

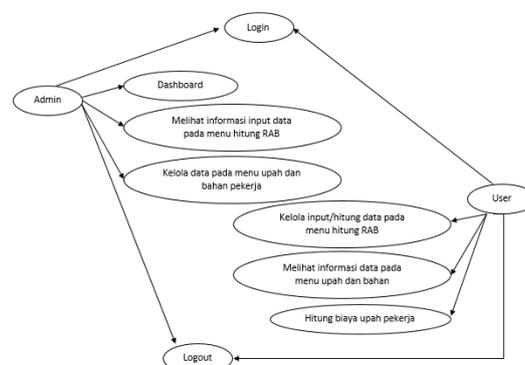
Pada tahapan pembuatan *System Informasi Perhitungan Rencana Anggaran Biaya Perencanaan Bangunan Berbasis Web* ini penulis menggunakan bantuan sebuah *platform codeigniter 3* untuk memudahkan pembuatan aplikasi dengan Bahasa pemrograman php sebagai pelengkap yang dinamis. Bahasa pemrograman php (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman yang berfungsi untuk mengolah data pada sebuah *local web server* [3]. *Codeigniter* merupakan sebuah *framework* atau kerangka kerja yang sifatnya *open source* dan biasanya digunakan untuk mengembangkan sebuah *system informasi (aplikasi) berbasis web*. *Codeigniter* banyak digunakan untuk mempermudah dalam pembuatan dan pengembangan *system informasi berbasis web* karena banyak menyediakan banyak jenis *library*. *Codeigniter* dibangun dan dikembangkan menggunakan konsep MVC yang dapat mengelola dan mengontrol data dari *database* yang diolah pada *controller* dan ditampilkan pada halaman *user interface* secara langsung [4]. Saat ini *codeigniter* telah diperbaharui menjadi *codeigniter* versi 4 dan untuk tahap proses membangun *system* aplikasi banyak berubah dan penulis belum menguasai *system* perancangan tersebut, karenanya penulis membangun sebuah *System Informasi Perhitungan Rencana Anggaran Biaya Perencanaan Bangunan Berbasis Web* dengan menggunakan *codeigniter* versi 3 dengan maksud dapat membuat *system* aplikasi dengan cepat dan sudah pasti *system* pembuatannya telah diketahui tanpa harus mempelajari *system* perencanaan membangun *system* aplikasi yang baru.

Penggabungan antara *system informasi (aplikasi)* yang dibangun dengan menggunakan bantuan sebuah *template framework codeigniter* dengan Bahasa pemrograman php sebagai pelengkap membangun suatu program *syntax*, dapat digunakan untuk membangun *System Informasi Perhitungan Rencana Anggaran Biaya Perencanaan Bangunan Berbasis Web*. Untuk menyimpan dan mengelola *input* data, penulis menggunakan *phpMyAdmin* sebagai wadah penyimpanan *database* yang dapat diakses melalui *software XAMPP Version: 8.0.10*. Penulis berharap dengan penelitian *System Informasi Perhitungan Rencana Anggaran Biaya Perencanaan Bangunan Berbasis Web* ini dapat mempermudah mengelola masukkan data pengguna yang memang membutuhkan *system informasi (aplikasi)* ini. *System* aplikasi ini dapat digunakan untuk mempermudah pengguna dalam mengelola *input* data secara otomatis, mencetak hasil data yang telah dikelola sebelumnya, dan juga dapat menghitung secara cepat dan otomatis pada *system* komputer.

## II. METODE PENELITIAN

### Use Case Diagram

*Use case diagram* digunakan sebagai gambaran alur kerja dari sistem secara sederhana. Adapun alur kerjanya terdapat berbagai fungsi utama pada sistem dan jenis pengguna yang saling berinteraksi pada masing-masing bagian sistem pemangilan [5].



Gambar 1. Use Case Diagram

Pada gambar 1 diatas menjelaskan tentang gambaran alur kerja dari *System Informasi Perhitungan Rencana Anggaran Biaya Perencanaan Bangunan Berbasis Web* yang digunakan sesuai kebutuhan pengguna dari *admin* maupun pengguna *user*. Hak akses *admin* dan *user* tidak di akses secara bersamaan, karenanya memiliki kebutuhan dan tujuan yang berbeda-beda sesuai masing-masing pengguna tersebut.

*Class Diagram*

Class diagram merupakan gambaran struktur *class* yang terdiri dari *id*, nama, atribut dan *role id* / kode pemanggilan *primary id* pada setiap tabel agar bisa saling terhubung. Terdapat 7 (tujuh) *class* yang dirancang dan digunakan pada sistem aplikasi untuk dikelola sesuai kebutuhan sistem [6].

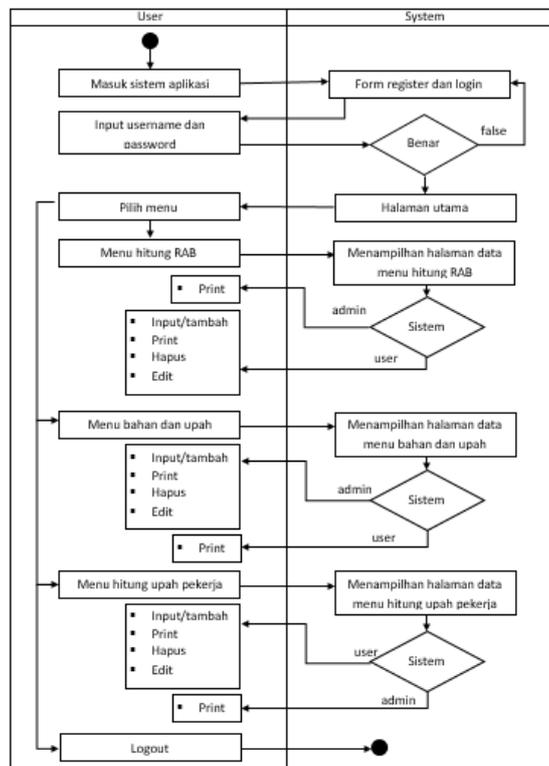


Gambar 2. Class Diagram

Pada gambar 2 diatas dapat dilihat secara jelas berisi kumpulan *table* yang digunakan dalam mengelola data pada *system* aplikasi. Table-table tersebut memiliki jenis *variable* kolom dan jenis tipe/*id* yang berbeda beda, agar dapat membedakan dan mempermudah proses pemanggilan *database* (*query* pada *id*) yang dibutuhkan sistem untuk dikelola *controller* sehingga dapat di tampilan pada *view* / *user interface* secara langsung.

*Activity Diagram*

Alur jalannya kegiatan sistem aplikasi untuk *admin* dan *user*/pengguna pada saat membuka *system* aplikasi hingga *function logout* sistem aplikasi *web*. Tahap awal alur sistem adalah saat membuka sistem aplikasi ini yaitu tampilan awal pada halaman *form login*. Apabila sistem meminta masukkan data pengguna yang sesuai dan pengguna tidak memenuhi syarat pada data yang ada, maka sistem akan memproses data sesuai permintaan pengguna.



Gambar 3. Activity Diagram.

Alur proses jalannya *activity diagram* pada gambar 3 diatas terdapat alur proses dalam menggunakan sistem aplikasi pada tahap awal membuka sistem kemudian saat pengguna melakukan *register* atau saat *login* hingga pengguna telah selesai *logout* pada sistem.

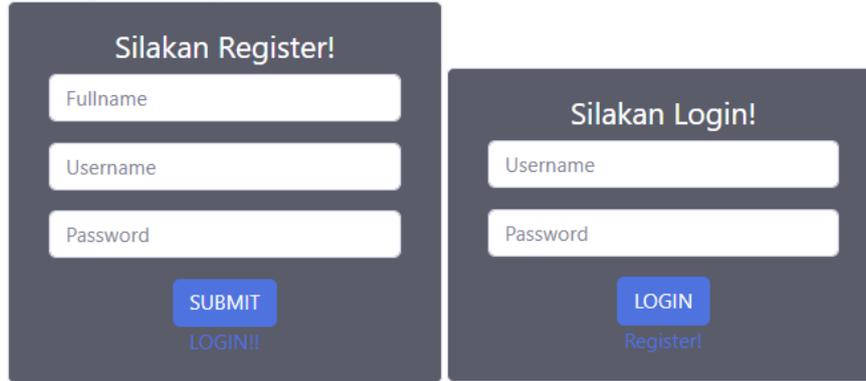
### III. HASIL DAN PEMBAHASAN

#### A. Implementasi Sistem

Berikut merupakan implementasi dari sistem aplikasi perhitungan RAB:

##### 1. From Register & Form Login

*Form register* ditampilkan apabila pengguna belum memiliki akun untuk membuka sistem aplikasi. Pada halaman *form register* akan terdaftar sebagai pengguna *user*. Sedangkan *form login* yang digunakan untuk masuk kedalam sistem aplikasi, dengan syarat telah terdaftar terlebih dahulu sebagai pengguna *admin* maupun *user*. Tampilan *form login* ini tampil saat awal membuka *system* aplikasi [7].

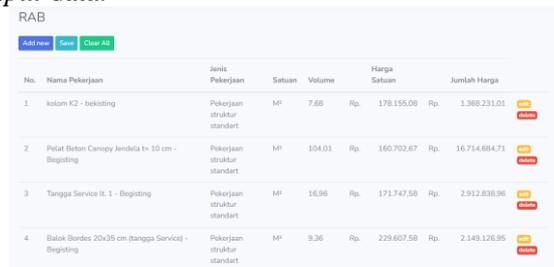


Gambar 4. *Form Register & Form Login*

Halaman *form login* dan *form register* pada gambar 4 diatas berfungsi untuk memasukkan data berupa nama, *username* dan *password* dengan tujuan agar pengguna dapat masuk ke dalam *system*. Pada saat pengguna *admin* atau *user* telah memasukkan *username* dan *password* pada *form login*, *system* akan memeriksa identitas dari *username* tersebut apakah data yang dimasukkan berperan sebagai pengguna *admin* atau *user*. Kemudian *system* akan menampilkan halaman utama masing-masing pengguna sesuai status pengguna itu apakah *admin* atau *user*.

##### 2. Menu Hitung RAB

*Form menu* Hitung RAB digunakan untuk mengelola *input* data, *edit* data, *delete* data maupun menyimpan/mencetak hasil *input* data.



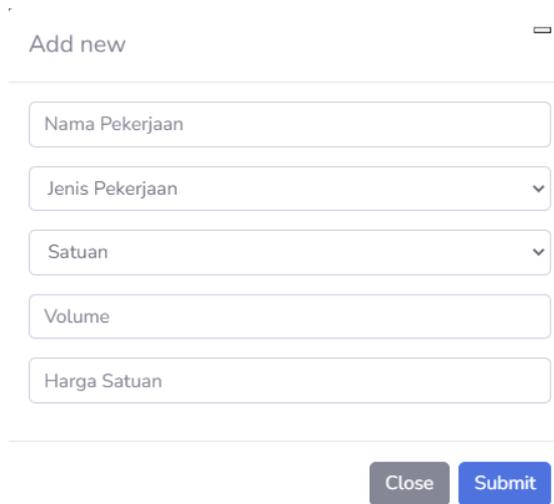
No	Nama Pekerjaan	Jenis Pekerjaan	Satuan	Volume	Harga Satuan	Jumlah Harga
1	Kolom K2 - bekisting	Pekerjaan struktur standart	MP	7,88	Rp. 178.155,08	Rp. 1.388.231,01
2	Pelat Beton Campur Jendala 1x 10 cm - Begisting	Pekerjaan struktur standart	MP	104,01	Rp. 160.792,67	Rp. 16.714.684,71
3	Tangga Service II - 1 - Begisting	Pekerjaan struktur standart	MP	18,36	Rp. 171.747,58	Rp. 2.922.838,96
4	Balok Bordes 20x35 cm (tangga Service) - Begisting	Pekerjaan struktur standart	MP	9,36	Rp. 229.607,58	Rp. 2.149.126,95

Gambar 5. *Page Menu Hitung RAB*

Gambar 5 diatas menunjukkan *view* halaman *menu* RAB. Halaman *menu* tersebut digunakan pengguna *user* untuk dapat mengelola data berupa *input*, *edit*, *delete*, *save/print document*. Data yang dimasukkan berupa keterangan nama atau deskripsi dari uraian pekerjaan, jenis kategori berdasarkan pekerjaan apa yang digunakan, input data volume dari keterangan pekerjaan berupa nilai angka beserta satuan volume (m, m<sup>2</sup>, m<sup>3</sup>, unit, dll) sesuai kebutuhan. Untuk nilai harga satuan upah pekerja, bahan material, dan alat yang akan dibutuhkan dapat diperiksa pada menu “Info harga upah dan bahan” pada *system* aplikasi. Dan pada *table* jumlah harga diatas di dapatkan dari hasil perkalian antara nilai volume dan harga satuan yang akan dibutuhkan.

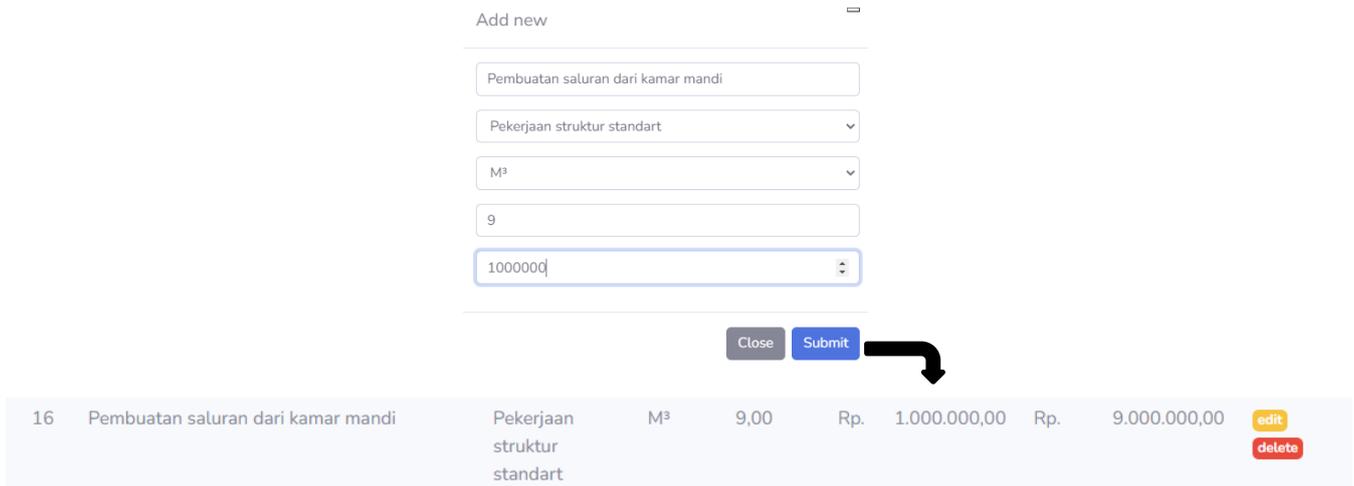
##### a) *Button Add new*

*Button add new* digunakan untuk memasukkan/*input* data sesuai dengan nama pekerjaan, jenis kategori, satuan, volume dan harga upah berdasarkan kebutuhan pengguna.



Gambar 6. Page Add New

Gambar 6 menunjukkan *view Add New modal system* aplikasi pada *menu RAB* diatas untuk memasukkan *input data*. Setelah memasukkan data pada *modal system* diatas, data akan di proses kemudian akan di tampilkan kembali pada halaman awal *menu RAB* dengan gambar 7 dibawah ini.

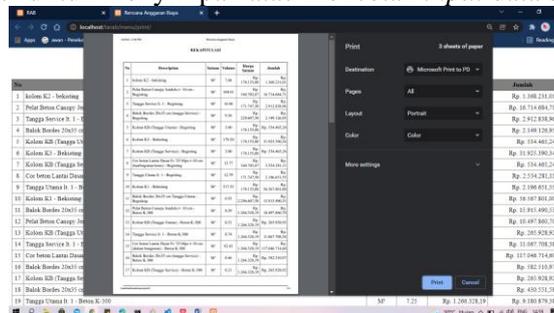


No	Uraian	Satuan	Volume	Harga Satuan	Jumlah
16	Pembuatan saluran dari kamar mandi	M <sup>3</sup>	9,00	Rp. 1.000.000,00	Rp. 9.000.000,00

Gambar 7. Tampilan Proses Add new Data dan Hasil Input Data

b) *Button Save*

*Button save* digunakan untuk menyimpan atau mencetak *input data* sesuai keinginan pengguna (*user*).

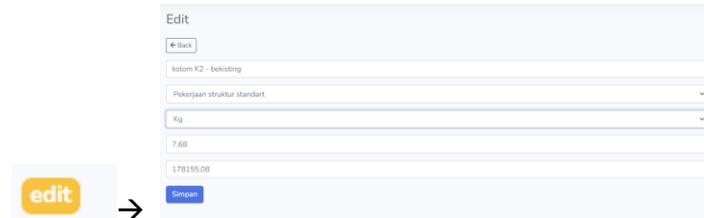


Gambar 8. Page Button Save/Print Pada Menu RAB

Gambar 8 diatas merupakan tampilan *view* yang berfungsi untuk menyimpan/mencetak data *table* yang telah di *input* sebelumnya oleh pengguna *user*. Pada penyimpanan data *table*-nya berupa dokumen dengan format pdf.

c) *Button Edit*

*Button edit* digunakan untuk mengubah data dari hasil *input* data sebelumnya sesuai dengan keinginan pengguna, juga dilakukan ketika saat *input* data tidak sesuai atau salah memasukkan data agar data yang dimasukkan sesuai.



Gambar 9. Page Button Edit

*Page button edit* ditampilkan pada gambar 9 diatas. *Form* edit diatas berfungsi untuk mengubah data sebelumnya apabila data tersebut mengalami suatu kesalahan pada tahap awal proses *input* data, kemudian pengguna ingin mengubah data tersebut dapat dilakukan di halaman ini.

d) *Button Delete*

→ *Button delete* berfungsi untuk menghapus data apabila data yang telah dimasukkan tidak digunakan atau data tidak sesuai/salah.

B. *Testing* (pengujian)

Adapun pengujian sistem aplikasi ini dilakukan dengan tahapan sebagai berikut.

1. Pengujian Sistem *Offline Browser*

Pengujian sistem ini dilakukan dengan cara mengakses *web* dengan menggunakan berbagai *browser*. Berikut ini merupakan tabel pengujian aplikasi *browser*.

TABEL I  
HASIL UJI DARI LOCAL WEB BROWSER

Web Browser	Hasil
Microsoft Edge .Ink	Berhasil
Google Chrome .Ink	Berhasil

Dari pengujian table 1 diatas bahwa *system* aplikasi dapat diakses menggunakan *browser Google Chome* dan *browser Microsoft Edge*. *Web browser* tersebut merupakan *web browser* yang sering digunakan untuk uji sistem setelah *coding*.

2. Pengujian *Usability*

*Usability* secara umum digunakan untuk menguji dengan mendapatkan respon berupa berhasil (*valid*) atau respon berupa kegagalan dalam penggunaan *system* aplikasi. Pengujian ini dilakukan melalui *form* pertanyaan *kuesioner* yang dapat mengolah data dengan *learnability*, *flexibility*, *effectivitiness*, dan *attitude*. Metode *kuesioner* berguna untuk mengukur kepuasan terhadap *system* aplikasi [8].

a) *Kuesioner*

Pengujian *kuesioner* dilakukan dengan *form* pertanyaan dan menyebarkan *kuesioner* yang dapat menghasilkan penilaian dari *responden* pada penggunaan *system* aplikasi. Jumlah *responden* berjumlah 16 pengguna [9].

TABEL II  
PERTANYAAN FORM KUESIONER

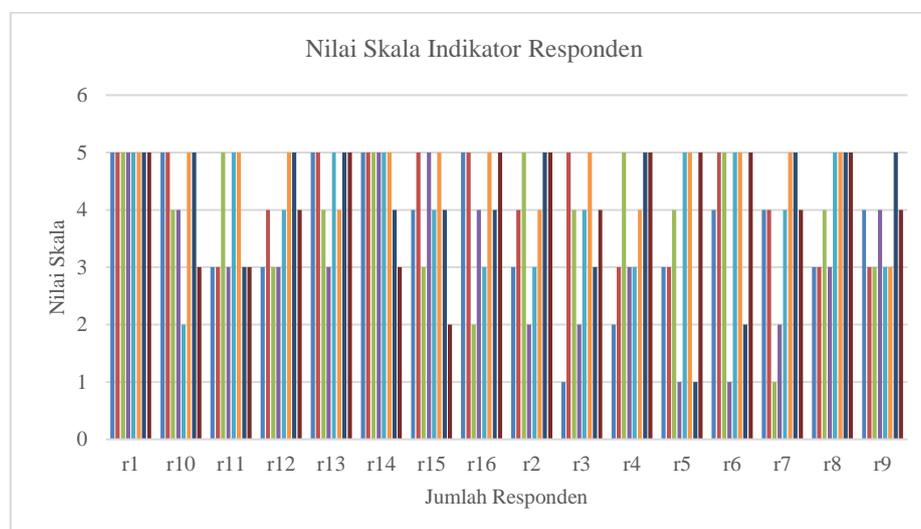
No	Indikator Pertanyaan	Kode
1	Apakah sistem aplikasi yang telah dibuat sesuai dengan kebutuhan pengguna?	P1
2	Apakah sistem aplikasi yang telah dibuat dapat berjalan dengan baik saat digunakan oleh pengguna?	P2
3	Apakah tampilan layout pada desain sistem aplikasi mudah untuk digunakan oleh pengguna?	P3
4	Untuk pilihan menu yang disediakan pada sistem aplikasi apakah sesuai dengan kebutuhan pengguna?	P4
5	Apakah pengguna mengalami kemudahan saat memasukkan data pada system aplikasi?	P5
6	Apakah system aplikasi yang telah dibuat dapat membantu mempermudah pengguna dalam menyelesaikan suatu permasalahan?	P6
7	Apakah setiap fitur tambahan pada system aplikasi yang telah dibuat berfungsi dengan baik sesuai tugasnya masing-masing?	P7
8	Apakah hasil dari masukkan data yang telah ditambahkan oleh pengguna sudah termasuk akurat?	P8

Pada Tabel II diatas terdapat 8 (delapan) pertanyaan *kuesioner* yang akan ditujukan untuk para *responden*/pengguna system aplikasi RAB untuk dapat memberikan penilaian terhadap *system* aplikasi yang telah dibuat. Untuk mendapatkan nilai rata-rata pemetaan yang digunakan adalah hasil dari nilai bobot pertanyaan *kuesioner* yang *respondennya* berjumlah 16 pengguna.

TABEL III  
NILAI SKOR DAN INTERPRETASI

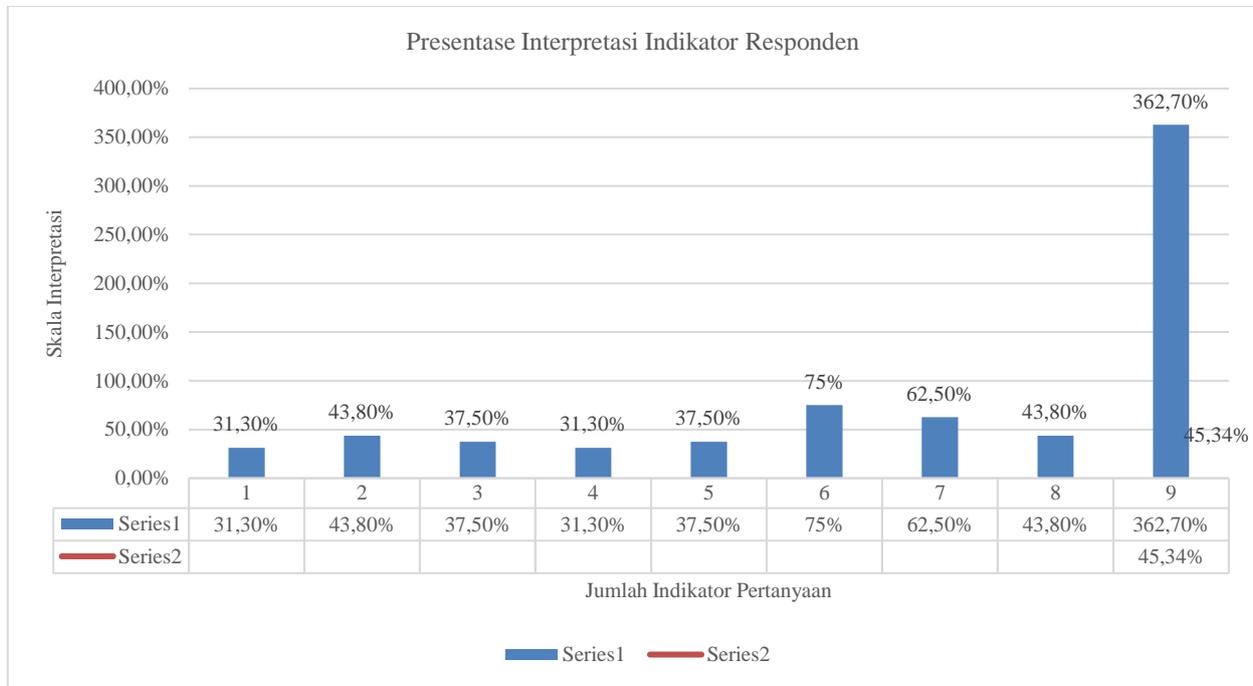
Interpretasi	Skor	Kode	Keterangan
0% - 20%	1	STS	Sangat Tidak Setuju
21% - 40%	2	TS	Tidak Setuju
41% - 60%	3	N	Netral
61% - 80%	4	S	Setuju
81% - 100%	5	SS	Sangat Setuju

Pada Tabel III diatas berisikan nilai bobot dengan besar nilai skala dan nilai *interpretasi* yang telah di tentukan didalam table tersebut. Dengan nilai skala yang tersebut, *responden* dapat menentukan nilai fungsi *system* aplikasi pada indikator pertanyaan yang telah disediakan sebelumnya pada *form* *kuesioner*.



Gambar 10. Grafik Presentase Penilaian Skala Responden

Gambar 10 diatas adalah menunjukkan sebuah gambar grafik hasil dari nilai *presentase* para pengguna/*responden* yang telah memberikan penilaian dari nilai skala tertentu terhadap *system* aplikasi yang telah dibuat sebelumnya. *Responden*/pengguna berjumlah 11 orang telah memberikan penilaian dengan skala 1-5 pada 8 (delapan) indikator pertanyaan.



Gambar 11. Grafik *Presentase Interpretasi Responden*

Gambar 11 diatas dapat kita lihat dengan nilai *presentase* yang tertera dengan jelas. Indikator pertanyaan yang diberikan pada *form kuesioner* memiliki 8 (delapan) buah pertanyaan yang menyangkut tentang bagaimana penilaian *system* aplikasi berjalan dengan baik atau tidak dengan memberikan nilai skala antara 1-5 [10]. Skala 1-5 tersebut berisi keterangan 1=sangat tidak setuju, 2=tidak setuju, 3=netral, 4=setuju, dan 5=sangat setuju. Dengan begitu untuk menghitung nilai rata-rata dapat diperoleh dari rumus:

$$\text{Hasil penilaian} = \frac{\text{Bobot} \times \text{Jumlah Responden}}{\text{Jumlah pertanyaan} \times \text{Total Jumlah Responden}}$$

Dengan begitu dapat disimpulkan pada gambar 11 diatas pada kode p1 memiliki nilai *presentase* sebesar 31,30%, p2 memiliki nilai *presentase* sebesar 43,80% hingga pada p8 memiliki nilai *presentase* sebesar 43,80%. Dan untuk nilai *presentase* rata-rata yang didapatkan sebesar 45,34%.

### III. KESIMPULAN

Secara umum *system* aplikasi RAB penelitian ini dibuat untuk menghitung secara cepat dan lebih efisien dalam menghitung jumlah total anggaran biaya dengan menampilkan rincian harga sesuai input/masukkan data oleh pengguna. *System* aplikasi RAB ini dirancang menggunakan sebuah *template framework codeigniter* versi 3 dengan *phpMyAdmin* sebagai *library* pengelolaan *database* dan untuk Bahasa pemrograman yang digunakan adalah Bahasa pemrograman php (*Hypertext Preprocessor*). Bahasa pemrograman php biasanya digunakan untuk membangun *system* berbasis *website*.

Dan untuk saran yang diberikan dari para *responden*, penulis merangkum sebagian saran tersebut yakni: untuk penggunaan *system* aplikasi masih belum begitu sempurna dan lebih baik diperbaiki dan lebih ditingkatkan lagi, tampilan masih belum begitu bisa dimanfaatkan karena tampilannya agak berbeda dengan *format* yang ada saat ini, masih di perlukan pengecekan terhadap hasil dari *subtitusi*, sangat membantu dalam hal *subtitusi*, mungkin bisa lebih di permudah dalam penggunaannya, untuk kebutuhan saat ini mungkin sudah cukup hanya masih perlu di kembangkan agar tidak 2 kali kerja.

## REFERENSI

- [1] Munzil Asri, Jafar Siddik & Supardin, “TINJAUAN RENCANA ANGGARAN BIAYA DAN WAKTU PELAKSANAAN PADA PROYEK PENINGKATAN JALAN MATANG BEN-PULO BLANG KABUPATEN ACEH UTARA”, JURNAL SIPIL SAINS TERAPAN VOLUME 02 NOMOR 02 September 2019.
- [2] Napu I, Wikarsa L, Suwanto T, “Aplikasi Perhitungan Rencana Anggaran Biaya Pembuatan Rumah Tinggal Berbasis Web”, JURNAL REALTECH
- [3] Arief Herdiansah, “System Development for Learning Process Monitoring in Private Lesson Institution Using Codeigniter Framework”, JISA (Jurnal Informatika dan Sains), Vol. 04, No. 01, June 2021.
- [4] Robby Yuli Endra, Ahmad Cucus & M. Aditya Wulandana S, “Perancangan Aplikasi Berbasis Web Pada System Aeroponik untuk Monitoring Nutrisi Menggunakan Framework Codeigniter”, Jurnal Sistem Informasi dan Telematika (Telekomunikasi, Multimedia dan Informatika), Vol.11 no.1|Juni 2020.
- [5] Achmad Fikri Sallaby & Indra Kanedi, “Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter”, Jurnal Media Infotama Volume. 16 Nomor. 1 Februari 2020.
- [6] A Subari, S Manan and E Ariyanto, “Design of E-office system in vocational school Diponegoro University using code igniter framework”, IOP Conf. Series: Materials Science and Engineering 801 (2020).
- [7] Suci Imani Putri & Yudistira Arya S, “Perancangan Sistem Perpustakaan Online Menggunakan Metode Model View Controller (MVC) Studi Kasus STMIK Asia Malang”, Jurnal Ilmiah Teknologi dan Informasia ASIA (JITIKA) Vol.10, No.2, Agustus 2016, ISSN: 0852-730X.
- [8] Yeni Anggraini, Donaya Pasha, Damayanti & Aan Setiawan, “SISTEM INFORMASI PENJUALAN SEPEDA BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS : ORBIT STATION)”, Jurnal Teknologi dan Sistem Informasi (JTSI) Vol. 1, No. 2, Desember 2020, 64 - 70 E-ISSN: 2746-3699.
- [9] I Wayan Darma Yasa, DKK, “Framework Codeigniter pada Rancang Bangun Prili (Primakara Library)”, Jurnal Ilmiah Ilmu Terapan Universitas Jambi, Volume 4 Nomor 2 Desember 2020.
- [10] Abdul Mubarak, “RANCANG BANGUN APLIKASI WEB SEKOLAH MENGGUNAKAN UML (UNIFIED MODELING LANGUAGE) DAN BAHASA PEMROGRAMAN PHP (PHP HYPERTEXT PREPROCESSOR) BERORIENTASI OBJEK”, JIKO (Jurnal Informatika dan Komputer) Ternate, Vol. 02 No. 1, April 2019.