**ANALISIS FITUR COSTUMER ADMINISTRASI**

**PADA APLIKASI ANTRIAN BERBASIS WEB**

**Dosen Pembina:** M. Yusril Helmi Setyawan, S.Kom., M.Kom.

Diajukan untuk memenuhi kelulusan matakuliah Proyek I

Program Studi DIV Teknik Informatika

**Oleh**

**AKMAL RESTU DEWANTORO (1.19.4.003)**

**M. RIZKY (1.19.4.021)**



**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**BANDUNG**

**2020**

**LEMBAR PERSETUJUAN**

**ANALISIS FITUR COSTUMER ADMINISTRASI**

**PADA APLIKASI ANTRIAN BERBASIS WEB**

**LAPORAN PROYEK I**

Program Studi DIV Teknik Informatika

**Oleh**

**AKMAL RESTU DEWANTORO (1.19.4.003)**

**M. RIZKY (1.19.4.021)**

Telah disetujui dan disarankan

Di Bandung pada tanggal \_\_\_\_\_\_\_\_\_\_\_\_\_

**Pembimbing**

M. Yusril Helmi Setyawan, S.Kom., M.Kom.

NIK 113.74.163

**SURAT PERNYATAAN TIDAK PLAGIAT**

Bandung , 02 April 2020

Sodara yang bertanda tangan dibawah ini :

Ketua Kelompok : M. Rizky (1.19.4.021)

Anggota : Akmal Restu Dewantoro (1.19.4.003)

Judul Laporan : ANALISIS FITUR COSTUMER ADMINISTRASI PADA

APLIKASI ANTRIAN BERBASIS WEB

Menyatakan bahwa kami tidak melakukan tindakan peniru, menjiplak ataupun menyalin skripsi karya ilmiah yang telah ada. Apabila kita terbukti melakukan tindakan tersebut , maka kita bersedia menerima sanksi yang diberikan sesuai dengan ketentuan yang ditetapkan dan berlaku di Program Studi Diploma IV Teknik Informatika Politeknik Pos Indonesia.

Mengetahui,

Anggota Kelompok

Akmal Restu Dewantoro

NPM. 1.19.4.003

Mengetahui,

Ketua Kelompok

M. Rizky

NPM. 1.19.4.021

**ABSTRAK**

Pada jaman sekarang ini, hampir semua kegiatan yang bersifat publik seperti contohnya administrasi pada bank mengharuskan nasabahnya agar mengantri untuk megurus semua keperluannya. Hal ini dikarenakan banyaknya nasabah yang dating untuk mengurus hal yang sama sehingga jika nasabah tidak tertib untuk mengantri maka proses pengurusan data nasabah akan sangat lama.

Seiring berjalannya waktu, aplikasi untuk antrian pun dibuat untuk mengatasi masalah – masalah tersebut. Namun dari banyaknya fitur pada aplikasi antrian berbasis web tersebut ternyata masih ada kekurangan di dalamnya , seperti misalnya tidak adanya fitur dimana nasabah dapat membatalkan proses antriannya, agar dapat mengefesienkan waktu petugas pada saat pemanggilan nomor antrian.

Fokus pada laporan ini adalah untuk menganalisis bagaimana fitur costumer administrasi pada aplikasi antrian berbasis web ini bekerja dan bagaimana menerapkan fitur untuk pembatalan nomor antrian agar dapan mengefesiensikan wanktu dari petugas.

Kata kunci : Analisis, Antrian, Efesiensi Waktu , Pembatalan.

*ABSTRACT*

*In this day and age, almost all public activities such as for example the administration of a bank that requests to queue to take care of all their needs. This relates to the income that comes to regulate the same thing so that it must be paid in an orderly manner to not wait in line so the data processing process will be very long.*

*Over time, an application for the queue was made to overcome these problems. However, from the number of features in this web-based queuing application, there are still shortcomings in it, as there are no features that can support the queuing process, in order to streamline the officer's time when calling a queue number.*

*The focus of this report is to analyze how customer features in this web-based queuing application work and use the feature to cancel the queue number so that it can streamline time from the clerk.*

*Keywords: Analysis , Queue , Time Efficiency , canceled.*

**KATA PENGANTAR**

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan Hidayah Nya sehingga kami dapat menyelesaikan laporan analisis yang berjudul “**ANALISIS FITUR COSTUMER ADMINISTRASI PADA APLIKASI ANTRIAN BERBASIS WEB**”.

Saya ucapkan terima kasih kepada bapak M. Yusril Helmi Setyawan, S.Kom., M.Kom. yang telah membantu kami baik secara moral maupun materi. Terima kasih juga saya ucapkan kepada teman seperjuangan saya yang telah mendukung kami sehingga kami dapat menyelesaikan tugas ini dengan tepat waktu.

Kami menyadari, bahwa laporan analisis yang kami buat ini masih jauh dari kata sempurna baik segi bahasa, penyusunan, maupun penulisannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang bisa membangun kami untuk mau berusaha lebih baik lagi dari sekarang, maupun dimasa mendatang.

Semoga laporan analisis ini bisa menambah wawasan para pembaca dan bisa bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan kita semua bahwasannya kita semua masih dalam proses pembelajaran.

Bandung, 2 April 2020

Penulis

**DAFTAR ISI**

-

**BAB 1**

**PENDAHULUAN**

* 1. **Latar Belakang**

Aplikasi pengantrian bank merupakan aplikasi yang berguna untuk mempermudah masyarakat untuk melakukan kegiatan seperti administrasi nasabah agar tetap tertib saat proses berlangsung. Nasabah hanya perlu mengambil nomor antrian dari aplikasi dengan menekan tombol antrian sesuai dengan keperluan nya dan aplikasi akan secara otomatis mencetak nomor antrian tersebut. Kemudian nasabah hanya perlu menunggu nomor antrian milik nya di panggil oleh petugas.

Namun belakangan ini sering terjadi kendala, ketika nasabah ingin mengurus keperluannya seperti mengurus administrasi atau mengurus transaksi, terkadang sering terjadi kendala dimana nasabah salah mencetak nomor antrian. Nasabah yang seharusnya mencetak nomor antrian untuk mengurus administrasi malah mencetak nomor antrian untuk mengurus transaksi karena unsur ketidak sengajaan. Ketika hal ini terjadi kebanyakan dari nasabah memilih untuk mengambil ulang nomor antrian yang sesuai dengan kebutuhannya. Tentu saja hal tersebut akan berdampak pada saat pemanggilan nomer antrian oleh petugas nantinya, salah satunya adalah waktu yang banyak terbuang karena nomor yang di panggil ternyata tidak ada yang memiliki.

Dengan adanya kendala tersebut, maka kami ingin merancang sebuah fitur yang dapat mengembalikan nomor antrian yang tidak terpakai tersebut agar pada saat pemanggilan oleh petugas kendala – kendala yang sudah di jelaskan tidak terjadi. Fitur *cansel* ini berupa barcode yang ketika di *scan* oleh aplikasi, nantinya secara otomatis nomor antrian tersebut akan di batalkan. Jadi petugas tidak perlu lagi untuk memanggil nomor antrian yang di batalkan tersebut.

* 1. **Identifikasi Masalah**

1. Seringnya terjadi kendala ketika nasabah ingin membatalkan nomor antrian nya dikarenakan salah print.
2. Sering terjadi kendala ketika nomor yang di panggil oleh petugas ternyata sudah dibatalkan.
3. Apa yang menyebabkan fitur *cansel* ini dibutuhkan khususnya pada aplikasi antrian.
4. Bagaimana menerapkan fitur *cansel* ini pada aplikasi antrian administrasi.
   1. **Tujuan**
5. Mengetahui bagaimana sistem dari aplikasi ini bekerja
6. Memahami algoritma dari sistem tersebut
7. Menerapkan fitur *cansel* pada aplikasi antrian khususnya pada antrian administrasi
   1. **Ruang Lingkup**

Agar perancangan aplikasi tersebut terkesan sederhana dan mudah di mengerti namun tidak mengurangi tujuannya, maka penulis membatasi perancangan aplikasi tersebut sebagai berikut:

1. Menganalisis sistem data costumer administrasi yang berjalan pada aplikasi antrian bank
2. Menganalisis sistem loket administrasi yang berjalan pada aplikasi antrian bank
3. Menganalisis manajemen akun
4. Menganalisis manajemen menu
5. Hanya membahas sampai bagaimana fitur *cansel* akan diterapkan pada aplikasi.
   1. **Sistematik Penulisan**

Sistematika penulisan dalam penyusunan laporan analisis sistem ini adalah sebagai berikut:

**BAB I PENDAHULUAN,** pada pada bab ini membahas tentang latarbelakang penulisan laporan dan pembuatan aplikasi , kemudian rumusan masalah yang membahas apa saja masalah yang terjadi pada aplikasi tersebut, kemudian tujuan yaitu membahas tentang apa saja yang ingin di capai apabila aplikasi ini telah selesai dalam pembuatan, kemudian ruang lingkup atau batasan masalah membahas tentang batasan penulis menganalisis aplikasi tersebut, terakhir adalah sistematik penulisan yang menjelaskan aturan tatapenulisan laporan ini.

**BAB II LANDASAN TEORI,** pada bab ini kita menjelaskan dari mana kita mendapatkan sumber-sumber untuk membuat aplikasi ini. Sumber yang digunakan haruslah valid dan tidak hasil analogi atau pendapat dari diri kita sendiri.

**BAB III ANALISIS DAN PERANCANGAN,** pada bab ini membahas tentang analisis sistem dari aplikasi tersebut , seperti membahas kekurangan dari aplikasi tersebut , bagaimana cara kerja dari sistem tersebut dan bagaimana perancangan nya.

**BAB IV IMPLEMENTASI DAN HASIL,** pada bab ini membahas tentang rincian komponen-komponen pendukung yang ada dalam aplikasi yang akan dibuat, dengan tatacara pamakaian , instalasi, serta pengujian perangkat lunak.

**BAB V KESIMPULAN DAN SARAN,** pada bab ini membahas tentang kesimpulan kita menganalisis aplikasi ini dan saran kedepanya bagi pembuat aplikasi jika tedapat kekurangan ataupun masalah agar kedepannya aplikasi dapat lebih baik lagi.

**BAB II**

**LANDASAN TEORI**

* 1. **Antrian**

Antrian merupakan prosedur dimana seseorang menunggu gilirannya untuk melakukan suatu kegiatan atau suatu keperluan. Proses antrian bertujuan untuk memaksimalkan efesiensi pelayanan terhadap permintaan.

* 1. **Pengertian *Database***

Menurut Fabbri dan Schwab, “Database merupakan sistem berkas terpadu yang dirancang untuk dapat meminimalkan pengulangan atau *redundancy* data*”.* maksud dari *redundancy* data tersebut adalah untuk mengurangi terjadinya inputan data yang ganda. Jadi pada dasarnya *database* digunakan sebagai tempat atau pusat penyimpanan data-data yang jumlahnya sangat bayak, namun data yang disimpan lebih *flexible* baik dari segi penyimpanan (*Storage*) ataupun dari segi manajemen data nya. Dengan adanya *database* maka tempat penampungan data akan jauh lebih efektif dan efisien.

* + 1. **MySQL**

MySQL merupakan sebuah *software DBMS* (*Database Management System*) yang berfungsi sebagai *server* / sistem untuk memanajemen data menggunakan *syntax SQL* agar data yang dikumpulkan dapat di kelompokan secara terstruktur. MySQL merupakan sebuah perangkat lunak gratis yang memiliki lisensi GPL ( *General Public License* ).

* + 1. **SQL**

SQL ( *Structured Query Language* ) Merupakan bahasa pemrograman yang digunakan untuk memanajemen data dalam RDBMS seperti contohnya MySQL. Perintah SQL sendiri sering disingkat dengan sebutan “*Query”.* SQL dibedakan menjadi 3 jenis diantaranya adalah :

1. Data Definition Language (DDL)

Jenis ini berkaitan dengan manipulasi tabel maupun database itusendiri, *Syntax* SQL yang digunakan diantaranya : CREATE , DROP , ALTER , dan RENAME.

1. Data Manipulation Language (DML)

Jenis ini berkaitan dengan manipulasi data pada suatu table di database seperti menghapus , mengubah , menambahkan dan membaca data / file. *Syntax* SQL yang digunakan diantaranya : SELECT, INSERT ,DELETE DAN UPDATE

1. Data Controll Language (DCL)

Jenis ini berkaitan dengan hak akses pengguna dapat masuk dan mengakses ke database. *Syntax* SQL yang digunakan diantaranaya:

GRAND dan REVOKE

* + 1. **XAMPP**

Xampp merupakan perangkat lunak gratis yang hampir dapat beroperasi di semua system operasi. Fungsi dari XAMPP ini adalah sebagai *sever* yaitu tempat menampung sebuah *database.* XAMPP terdiri dari beberapa program, diantaranya adalah *Apache HTTP Server, MySQL Database* , dan penerjemah bahasa yaitu *PHP* dan *Perl*. Nama XAMPP sendiri merupakan singkatan dari “X” yang artinya tempat sistem operasi ( *base* ), lalu “A” singkatan dari *Apache* , “M” singkatan dari *MySQL* , kemudian “PP” singkatan dari *PHP* dan *Perl.*

* 1. **PHP**

PHP ( *Hypertext Preprocessor* ) merupakan sebuah bahasa pemrograman yang umumnya digunakan untuk membangun website. Bahasa PHP populer karena *script* dari PHP bisa di sisipkan kedalam *html .* berikut ini adalah kode PHP yang sering/banyak digunaka :

1. ECHO

Merupakan *script* yang digunakan untuk menanpilkan data ke layar.

1. INCLUDE

Merupakan *script* yang digunakan untuk memanggil atau menyisipkan file.

1. MYSQL\_CONNECT

Merupakan *script* yang digunakan untuk melakukan koneksi ke database MySQL.

1. MYSQL\_QUERY

Merupakan *script* yang digunakan untuk menjalankan *Query* *SQL* pada MySQL.

1. HEADER

Merupakan *script* yang digunakan untuk mengubah tipe header pada suatu laman atau untuk *redirect* ( mengalihkan ) ke halaman lain.

1. IF STATEMENT

Merupakan *script* pengkondisian yang nilainya *Boolean* antara *TRUE* dan *FALSE*

1. WHILE

Merupakan *script Looping* atau pengulangan data, looping akan tetap berjalan jika kondisinya *TRUE* .

1. FOREACH

Merupakan *script Looping* atau pengulangan data, perbedaannya penggunaan *foreach* biasanya adalah untuk mengekstrak data yang di ambil dari database.

1. ARRAY

Merupakan *script* untuk merubah tipe data menjadi *array* . biasanya tipe data *array* dilambangkan dengan tanda kurung siku [ ] .

* + 1. **OOP**

OOP ( *Object Oriented Programing* ) merupakan konsep atau teknik dalam pemrograman modern agar kode yang dihasilkan lebih rapih , terstruktur, dan efisien metode ini banyak digunakan karena dapat mempermudah nantinya pada saat skala program yang sedang di kembangkan tersebut sudah semakin besar. Pada konsep OOP kita harus mengenali pondasi dasar dari konsep tersebut , diantaranya adalah :

1. CLASS

Class merupakan sebuah wadah atau *blue print* yang digunakan sebagai cetakan untuk membuat sebuah *object*. Dalam pemrograman atau dalam konsep OOP , sebuah *class* dapat berisikan *property* dan *method.*

1. OBJECT

Object sendiri merupakan implementasi atau bentuk hasil dari *class* yang di instansiasikan, ini berarti suatu *object* dapat menggunakan atau mengakses fungsi-fungsi pada *class* tersebut.

1. PROPERTY

Propery merupakan data – data yang menempel atau berada pada suatu *class* yang nantinya dapat digunakan atau dimanipulasi sesuai dengan kebutuhan.

1. METHOD

Method merupakan sebuah fungsi yang dibuat dan disimpan di dalam *class*, Method sendiri bisa juga disebut dengan prosedur ( tergantung tipe data *return* *method* tersebut ) selain itu *method* memiliki keterkaitan dengan *class* itu sendiri, sebagai contoh sebuah *class* dengan nama “mobil” seharusnya memiliki *method* seperti maju , mundur , berhenti dan lain sebagainya yang berkaitan dengan mobil.

* 1. **JAVASCRIPT**

*Javascript* merupakan sebuah bahasa pemrograman tingkat tinggi , artinya bahasa pemrograman ini mudah dimengerti oleh manusia. bahasa *javascript* hanya spesifik digunakan untuk aplikasi web dan *javascript* merupakan sebuah bahasa *scripting* yaitu bahasa pemrograman yang tertanam di dalam sebuah program yang lebih besar contohnya adalah bahasa pemrograman PHP. berikut ini adalah kode dan istilah – istilah pada *Javascript* yang sering/banyak digunakan :

1. JSON

JSON ( *Javascript Object Notation* ) merupakan sebuah konsep pemformatan data biasanya konsep ini digunakan untuk melakukan pertukaran dan pengimpanan data. Pemformatan pada *json* hampir mirip seperti konsep *array.*

1. AJAX

Ajax (*Asynchronous Javascript and XMLHTTP*) merupakan teknik pada *javascript* yang digunakan untuk melakukan interaksi atau pertukaran data dengan server, *script* akan berjalan secara *Asynchronous* dan berjalan di belakanga layartanpa harus mereset/menyegarkan laman yang sedang digunakan.

1. JQUERY

JQuery merupakan sebuah *framework* yang populer dari bahasa pemrograman *javasript*. tujuannya adalah untuk mempermudah penulisan kode dari javascript itu sendiri.

1. DATATABLE

DataTable merupakan sebuah *framework* yang populer untuk mempermudah pada saat pembuatan tabel untuk menampilkan data. Data Table menggunakan bahasa *Javascript* sebagai base nya.

1. TAG SCRIPT

Tag Script digunakan untuk menandakan bahwa kode didalamnya merupakan bagian dari *Javascript*. Contoh tag pembuka dan penutup script : <script></script>

**BAB III**

**ANALISIS DAN PERANCANGAN**

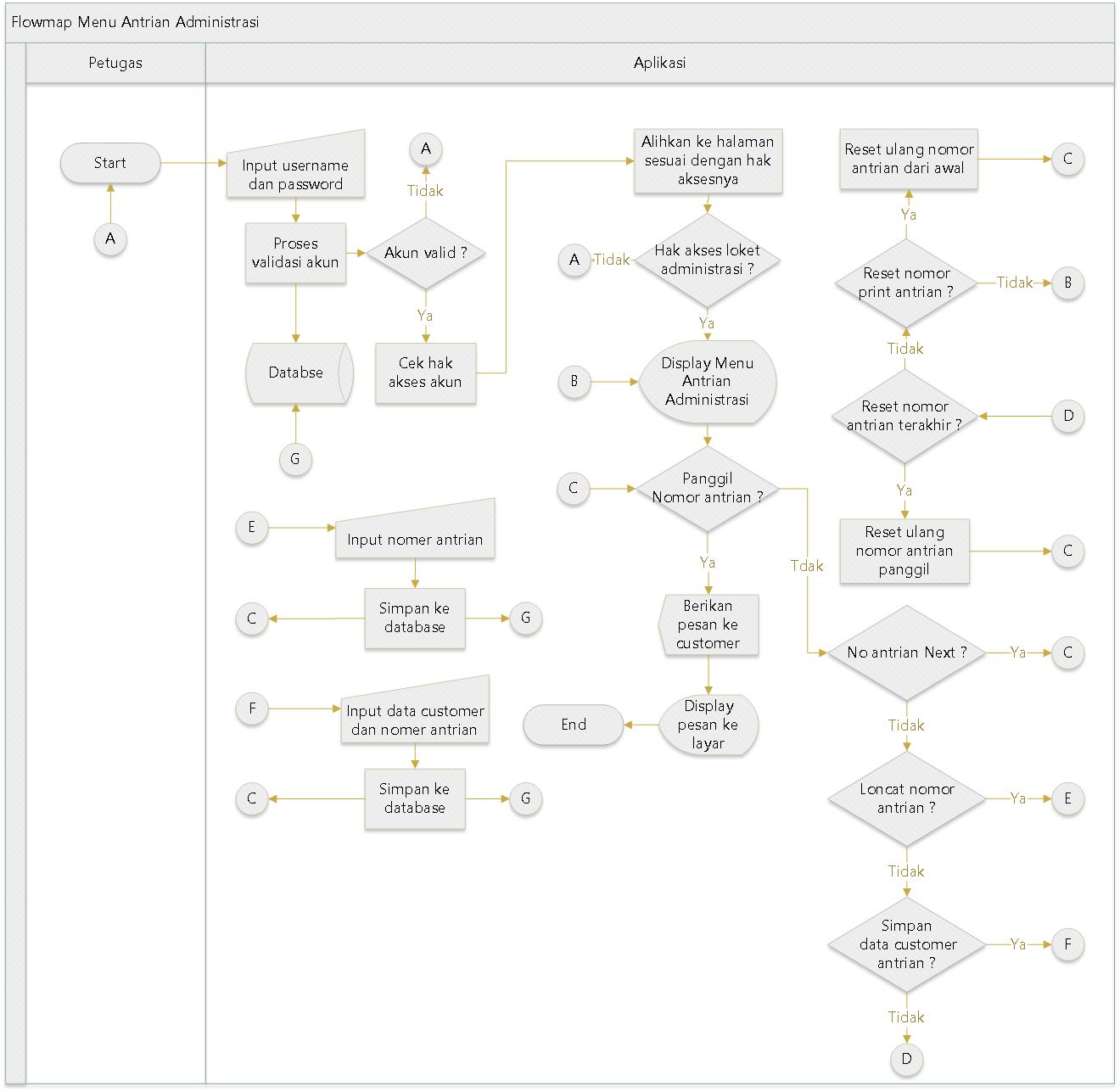
## **Analisis Sistem**

Analisis sistem merupakan kegiatan untuk menguraikan informasi dari suatu sistem dengan tujuan untuk mengidentifikasi serta mengevaluasi permasalahan – permasalahannya, yang kemudian diharapkan dapat di usulkan perbaikan – perbaikan nya. Bada bagian ini , akan membahas tentang bagan alur *flowmap* dari aplikasi/sistem antrian khususnya pada fitur *customer* administrasi, yang isinya mengenai analisis prosedur dan aliran dokumen yang sedang berjalan, kemudian membahas tentang pengkodean dan analisis sistem *non-fungsional* yang meliputi perangkat keras dan perangkat lunak yang di gunakan, serta analisis *user* yang terlibat di dalam aplikasi/sistem antrian pada fitur *customer* administrasi.

* + 1. **Analisis Sistem Berjalan ( *Current System* )**

Sistem yang berjalan saat ini terdiri dari beberapa prosedur, diantaranya adalah fitur *login* , fitur loket administrasi yang terdiri dari menu antrian dan data *customer* administrasi , fitur layar lcd untuk menampilkan nomor antrian yang di panggil oleh petugas ke user, dan fitur layar printer yang berguna untuk mencetak nomor antriannya.

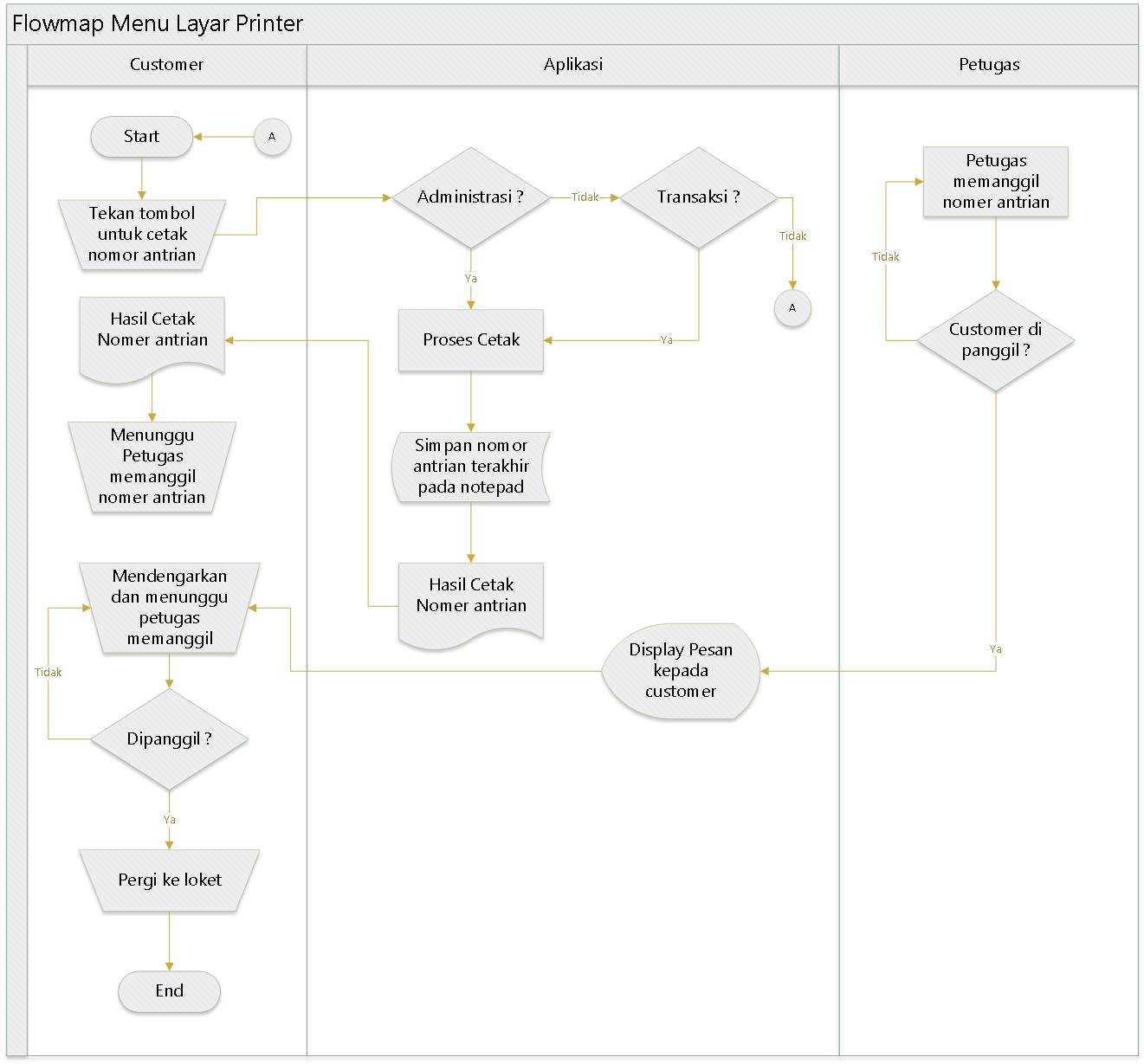
1. **Analisis Prosedur ( *flowmap* )**
2. **Analisis Sistem Yang Berjalan Pada Menu Antrian Administrasi**

Untuk menjalankan fitur pada menu antrian pengguna harus *login* terlebih dahulu dengan menggunakan akun yang memiliki hak akses pada menu tersebut, setelah login kemudian pengguna dapat mengakses menu antrian administrasi tersebut. Di dalam menu antrian administrasi terdapat beberapa fitur di antaranya adalah menu untuk pemanggilan nomor antrian dengan suara, menu *jump* antrian untuk loncat ke nomor antrian yang di inginkan, menu *save* antrian untuk simpan data antrian, menu *restart* untuk mereset nomor antrian ( kembali ke awal ), dan terakhir fitur reset print untuk mereset nomor antrian pada menu layar printer yang di ambil/dicetak oleh *customer*. Adapun *flowmap* prosedur menu antrian administrasi adalah sebagai berikut :

*Gambar 3.1 Flowmap analisis sistem yang berjalan pada prosedur menu antrian administrasi*

1. **Analisis Sistem Yang Berjalan Pada Menu Layar Printer**

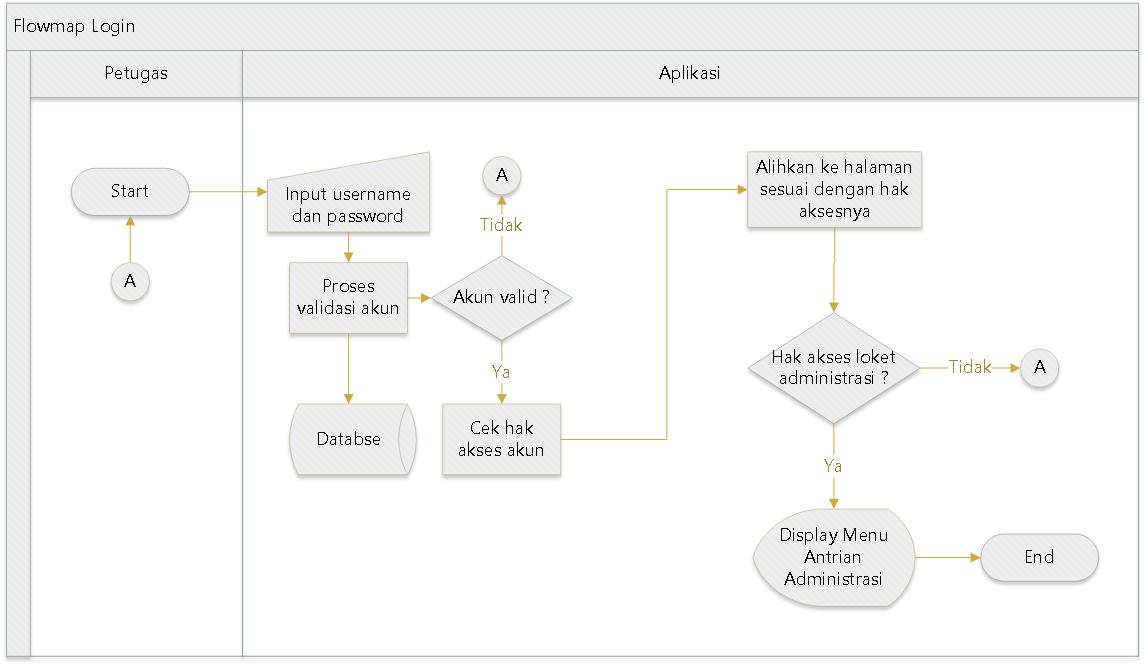
Pada prosedur pencetakan nomor antrian *customer* menekan tompol cetak sesuai dengan keperluannya, ada 2 pilihan yaitu untuk mengurus administrasi dan untuk mengurus transaksi, pada pembahasan ini hanya membahas untuk pengurusan administrasi. Setelah *customer*  mendapatkan nomor antriannya , kemudian *costumer* hanya tinggal menunggu petugas loket memanggil nomor antrian dan mengarahkan ke loket sesuai dengan keperluan *customer* tersebut. Adapun *flowmap* prosedur pada menu layar printer adalah sebagai berikut :



*Gambar 3.2 Flowmap analisis sistem yang berjalan pada prosedur layar printer.*

1. **Analisis Sistem Yang Berjalan Pada Prosedur Login**

Untuk dapat menjalankan aplikasi ini pengguna (petugas) harus *login* terlebih dahulu. Setelah *login* sesuai dengan hak aksesnya kemudian setelah itu petugas dapat mengoperasikan aplikasi. Adapun *flowmap* dari prosedur login adalah sebagai berikut :



*Gambar 3.3 Flowmap analisis sistem yang berjalan pada prosedur login.*

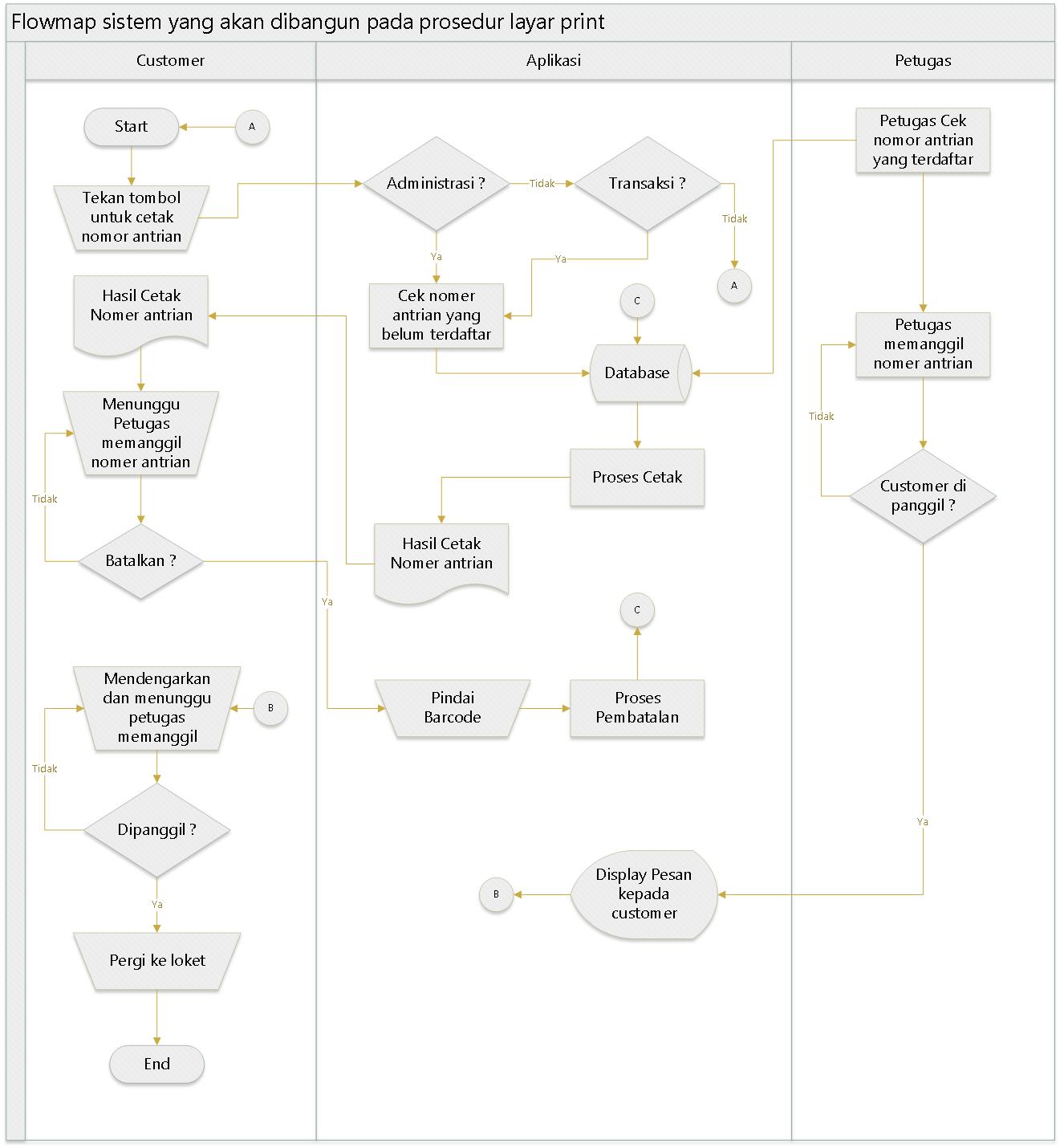
* + 1. **Analisis Sistem Yang Akan Dibangun**

Pada bagian ini berisikan *flowmap* dari sistem yang akan dibangun yaitu pada prosedur menu layar printer. Di sini kami menambahkan menu *cansel* untuk membatalkan nomor antrian agar dapat mengefesiensikan waktu dari petugas. Adapun *flowmap* yang akan dibangun adalah sebagai berikut :

1. **Analisis sistem yang akan dibangun pada prosedur layar printer.**

Pada prosedur layar printer melibatkan 3 entitas yaitu petugas , kemudian aplikasi antrian dan terakhir *customer* itu sendiri. *Customer* datang ke loket antrian kemudian menekan tombol pada layar aplikasi untuk mencetak nomor antrian sesuai dengan keperluannya, kemudian apabila *customer* ingin membatalkan nomor antriannya dikarenakan ada masalah seperti salah mencetak nomor atau jika *customer* ada keperluan mendadak dan ingin membatalkan nomor antriannya, *customer* dapat menggunakan fitur *cansel* untuk membatalkannya , sehingga dapat mengefesiensikan waktu dari petugas loket.

Fitur ini menggunakan *barcode* yang tercetak pada kertas nomor antrian, ketika customer ingin membatalkan nomor antriannya cukup dengan pindai *barcode* tersebut dengan aplikasi, maka secara otomastis nomor antrian akan tebatalkan dan nomor antrian yang dibatalkan dapat di gunakan kembali oleh *customer* lainnya. Adapun *flowmap* dari sistem yang akan dibangun adalah sebagai berikut:



*Gambar 3.4 Flowmap analisis sistem yang akan dibangun pada prosedur layar print.*

* + - 1. **Kebutuhan Fungsional (Functional Requirements)**

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan kebutuhan sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, Berikut proses yang terdapat pada Aplikasi Antrian :

1. Proses Login untuk mengelola siapa saja yang dapat menggunakan Aplikasi sesuai level yang telah ditentukan.
2. Proses untuk mengelola data Costumer. Diantaranya pengelolaan Penambahan data Costumer dan detail Costumer yang meliputi *input , update* dan *delete.*
3. Proses untuk menampilkan data Costumer yang telah di simpan.
4. Proses untuk mengelola data Akun. Diantaranya pengelolaan Penambahan data Akun dan detail Akun yang meliputi *input , update* dan *delete.*
5. Proses untuk menampilkan data Akun para pengguna Aplikasi.
6. Proses untuk mengelola Daftar Menu. Diantaranya pengelolaan Penambahan Daftar Menu yang akan di gunakan dalam aplikasi yang meliputi *input , update* dan *delete.*

**Kebutuhan Non-Fungsional (*Non- Functional Requirement*)**

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis pengguna/*User*. Adapun kebutuhan fungsional yang akan dibuat adalah sebagai berikut :

1. **Kebutuhan Perangkat Keras**

Tabel 3.3 Deskripsi Perangkat Keras *Server*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** | **Keterangan** |
| 1 | *Hardisk* | 320 GB | Media untuk menyimpan data aplikasi yang dibuat |
| 2 | *Memory* | 3 GB | *Memory System* yang digunakan |
| 3 | *Processor* | *Intel® core™ 2 duo CPU T6570 @2,10 Ghz* | 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 |

Tabel 3.4 Deskripsi Perangkat Keras *Client*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** | **Keterangan** |
| 1 | *Hardisk* | 160 GB | Sebagai tempat untuk menyimpan data yang dibutuhkan tetapi pada sisi client tidak diharuskan memiliki ketersediaan space yang besar |
| 2 | *Memory* | 1 GB | Kecepatan *client* dalam mengakses *system* ini |
| 3 | *Processor* | *Intel® core™ 2 duo CPU T6570 @2,10 Ghz* | Untuk perHalamanansi komputer |
| 4 | Infrastruktur Jaringan |  | *Server* dan *Host* |

1. **Kebutuhan Perangkat Lunak**

Tabel 3.5 Deskripsi Perangkat Lunak *Server*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tools / Software** | **Fungsi** | **Keterangan** |
| 1. | *Windows 8* | Sistem Operasi | **-** |
| 2. | *Xampp 3.2.1* | *Server* Basis Data | **-** |
| 3. | *PHP, JavaScript* | Bahasa Pemograman Yang digunakan | **-** |
| 4. | *Sublime Text, Visual Studio Code* | *Software* Pendukung | **-** |
| 5. | *PDF* | *Document* | **-** |
| 6. | *Google Crome* | *Browser* | **-** |

Tabel 3.6 Deskripsi Perangkat Lunak *Client*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tools / Software** | **Fungsi** | **Keterangan** |
| 1. | *Windows 7* | Sistem Operasi | **-** |
| 2. | *Xampp 3.2.1* | *Server* Basis Data | **-** |
| 3. | *PHP, JavaScript* | Bahasa Pemograman Yang digunakan | **-** |
| 4. | *Sublime Text, Visual Studio Code* | *Software* Pendukung | **-** |
| 5. | *PDF* | *Document* |  |
| 6. | *Google Crome* | *Browser* | **-** |

1. **Analisis Pengguna/*User***

Aplikasi yang akan digunakan memiliki fitur – fitur yang berbeda – beda seusai dengan fungsinya masing – masing. Akan tetapi untuk mengakses fitur – fitur tersebut harus memiliki akun yang telah di daftarkan terlebih dahulu dan juga setiap akun yang telah di daftarkan mempunyai fungsi yang berbeda juga sesuai *level*/tingkat hak akses yang telah ditentukan. Pada Aplikasi Antrian ini terdapat 4 *level*/tingkat hak akses yang tersedia yaitu, Admin, Administrator, Pas / Antrian Transaksi, dan Print. Untuk lebih jelasnya, Berikut penjelsannya :

1. User Admin

Admin disini sebagai super user yang dapat mengakses seluruh fitur pada aplikasi.

1. User Administrasi / loket administrasi

User disini sebagai pengurus antrian pada loket administrasi. user ini tidak dapat mengakses fitur pada menu antrian transaksi.

1. User pas / loket pas ( antrian transaksi )

User disini mirip seperti User Administrasi" namun user mengurus hal yang berbeda, yaitu mengurus antrian pada loket transaksi. User loket pas tidak dapat mengakses fitur dari menu antrian administrasi.

1. User Print

Berbeda dengan user lainnya, user print disini bertugas sebagai penanggung jawab antarmuka dengan nasabah / pelanggan yang ingin mengambil nomor antriannya pada menu "layar printer", user print tidak dapat mengakses fitur administrasi dan transaksi.

## 

## **Kamus Data**

Kamus data ( *Data Dictionary* ) merupakan sekumpulan data dan kebutuhan – kebutuhan informasi dari suatu sistem. Kamus data digunakan untuk mendefinisikan data yang mengalir pada sebuah sistem secara lengkap. Kamus data dibuat berdasarkan arus data yang ada pada diagram aliran data. Kamus data yang digunakan untuk membangun fitur customer administrasi pada aplikasi antrian berbasis web ini yaitu :

1. Kamus Data : Form Login Aplikasi

Nama Arus Data : Masuk ke Aplikasi

Alias : Form Login

Arus Data : Admin -> Pengisian Form Login, Pengisian Form Login -> Validasi, Validasi -> Menu Utama

Periode : Setiap Ingin Menggunakan Aplikasi

Penjelasan : Pengisian Form Login untuk masuk ke Aplikasi

Struktur Data :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Item Data** | | **Type** | **Lebar** | **Keterangan** |
| **Di Sistem** | **Di Program** |
| 1. | User Id | User\_Id | Int | 11 | ID |
| 2. | Username | Username | Varchar | 20 | Username |
| 3. | Password | Password | Varchar | 40 | Password |
| 4. | NIP | NIP | Varchar | 20 | NIP |
| 5. | Nama Lengkap | Nama\_Lengkap | Varchar | 50 | Nama Lengkap |
| 6. | Level | Level\_ID | TinyInt | 4 | Level Id |

1. Kamus Data : Antrian Nasabah

Nama Arus Data : Pengambilan No. Antrian Nasabah

Alias : No Antrian

Arus Data : Nasabah -> Proses ambil no antrian, Proses ambil no antrian -> No Antrian

Periode : Setiap kali nasabah yang ingin melakukan proses administrasi

Penjelasan : Pengambilan nomer antrian nasabah untuk melakukan proses

administrasi

Struktur Data :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Item Data** | | **Type** | **Lebar** | **Keterangan** |
| **Di Sistem** | **Di Program** |
| 1. | No Antrian | nomor\_antrian | Varchar | 255 | Nomer Antri Nasabah |
| 2. | Waktu Antrian | time | Varchar | 255 | Waktu No Antrian Terakhir |

1. Kamus Data : Simpan Antrian

Nama Arus Data : Pendaftaran No. Antrian Nasabah

Arus Data : Admin -> input data nasabah, input data nasabah -> Proses simpan antrian, Proses simpan antrian -> No. Antrian Tersimpan

Periode : Setiap nasabah yang memiliki keperluan lain selain proses administrasi.

Penjelasan : Pendaftaran nomer antrian nasabah yang memiliki keperluan lain selain proses administrasi.

Struktur Data :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Item Data** | | **Type** | **Lebar** | **Keterangan** |
| **Di Sistem** | **Di Program** |
| 1. | No Antrian | nomor\_antrian | Varchar | 50 | Nomer Antrian |
| 2. | Nama | nama | Varchar | 255 | Nama Nasabah |
| 3. | Jenis Kelamin | Jenis\_Kelamin | Varchar | 10 | Jenis Kelamin Nasabah |
| 4. | Nomer Telepon | No\_telpon | Varchar | 20 | Kontak Nasabah |

**DAFTAR PUSTAKA**

[1] https://id.wikipedia.org/wiki/MySQL

[2] <https://id.wikipedia.org/wiki/Pengadaan>

[3] <https://www.duniailkom.com/tutorial-mysql-pengertian-sql-structured-query-language/>

[4] <https://www.malasngoding.com/php-oop-part-2-pengertian-class-object-property-dan-method/>

[5] <https://id.wikipedia.org/wiki/PHP>

[6] <https://id.wikipedia.org/wiki/JavaScript>

[7] <https://id.wikipedia.org/wiki/AJAX>

[8] https://id.wikipedia.org/wiki/XAMPP

[9] A. J.Fabbri and A. R. Schwab, Practical Database Management, Boston: PWS-KENT, 1992.