DOUBLE LINKED LIST SIRCULAR**MEYKA YULIANTI****185100027**

Fakultas Komputer

meykayulianti.student@umitra.ac.id**Abstract**

Dalam pemakaian sehari-hari istilah senarai berantai (linked list) adalah kumpulan linear sejumlah data. Bantak contoh yang menunjukkan senarai berantai seperti pada daftar belanjaan, yang berupa barang pertama, kedua, ketiga dan seterusnya. Untuk hari berikutnya, maka daftar tersebut bisa berubah sesuai dengan barang yang harus dibeli lagi atau barang yang tidak perlu dibeli lagi.

Double Linked List Circular adalah linked list dengan menggunakan pointer, dimana setiap node memiliki 3 field, yaitu:

- 1 field pointer yang menunjuk pointer berikutnya (next),
- 1 field menunjuk pointer sebelumnya (prev),
- serta sebuah field yang berisi data untuk node tersebut.

Double Linked List Circular pointer next dan prev nya menunjuk ke dirinya sendiri secara circular. Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya.

Dibutuhkan dua buah variabel pointer yaitu head dan tail. Head akan selalu menunjuk pada node pertama, sedangkan tail akan selalu menunjuk pada node terakhir.

Kata Kunci : Double Linked List Circular

A. PENDAHULUAN

Pesatnya teknologi, terutama teknologi komputer sudah tak bisa dipungkiri lagi, bagi yang mengikuti perkembangannya, ia tidak akan dipandang sebelah mata. Sebaliknya, bagi yang tidak mengikuti perkembangannya, bersiaplah untuk mundur secara suka rela dari panggung kompetisi. Ibarat wabah, teknologi komputer sudah menyusupi hampir semua bidang kehidupan manusia. Dari pemerintah pusat sampai tingkat pemerintah desa, perusahaan-perusahaan, supermarket, minimarket, perguruan tinggi, SLTA, SLTP, bahkan SD hampir semuanya mengenal komputer.

Saat ini, mempunyai lingkungan yang semakin luas dan banyak diminati, juga dapat digunakan untuk menghasilkan uang adalah dunia pemrograman komputer. Dalam dunia pemrograman komputer, dikenal banyak bahasa pemrograman, seperti C, C++, Pascal, Basic, Java, dan lain-lain.

Jika dikembangkan dari berbagai bahasa pemrograman ini terdiri dari beberapa struktur data, dan diantaranya adalah linked list. Dalam pemakaian sehari-hari istilah senarai berantai (linked list) adalah kumpulan linear sejumlah data. Bantak contoh yang menunjukkan senarai berantai seperti pada daftar belanjaan, yang berupa barang pertama, kedua, ketiga dan seterusnya. Untuk hari berikutnya, maka daftar tersebut bisa berubah sesuai dengan barang yang harus dibeli lagi atau barang yang tidak perlu dibeli lagi.

B. PEMBAHASAN / STUDI KASUS

Double Linked List Circular

1. Double Linked List Circular

Double Linked List Circular adalah linked list dengan menggunakan pointer, dimana setiap node memiliki 3 field, yaitu:

- 1 field pointer yang menunjuk pointer berikutnya (next),
- 1 field menunjuk pointer sebelumnya (prev),
- serta sebuah field yang berisi data untuk node tersebut.

Double Linked List Circular pointer next dan prev nya menunjuk ke dirinya sendiri secara circular. Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya. Untuk pembentukan node baru, mulanya pointer next dan prev akan menunjuk ke dirinya sendiri. Jika sudah lebih dari satu node, maka pointer prev akan menunjuk ke node sebelumnya, dan pointer next akan menunjuk ke node sesudahnya.

Semua sel yang terdapat pada list disambungkan dengan pointer, sedangkan tiap sel memiliki tiga komponen yaitu value, pointer ke sel sebelumnya dan pointer ke sel berikutnya. Dengan memiliki dua buah pointer ini, maka doubly-linked list dapat diakses dengan dua arah, ke arah depan dan ke belakang.

Dibutuhkan dua buah variabel pointer yaitu head dan tail. Head akan selalu menunjuk pada node pertama, sedangkan tail akan selalu menunjuk pada node terakhir.

1. Inisialisasi Node

Manipulasi linked list tidak bisa dilakukan langsung ke node yang dituju, melainkan harus melalui node pertama dalam linked list.

Deklarasinya sebagai berikut:

```
TNode *head, *tail;
```

2. Fungsi Inisialisasi Double Linked List Circular

```
void init()
{
    head = NULL;
    tail = NULL;
}
```

3. Function isEmpty

Fungsi isEmpty digunakan untuk mengetahui kosong tidaknya DLLC. Berikut ini adalah pendeklarasian fungsi isEmpty :

```
int isEmpty()
{
    if(head == NULL) return 1;
    else return 0;
```

4. Penambahan Node di Depan

Penambahan node baru akan dikaitkan di node paling depan, namun pada saat pertama kali (data masih kosong), maka penambahan data dilakukan pada head nya dan node baru ini menempati posisi sebagai head dan tail.

Pada prinsipnya adalah mengkaitkan data baru dengan head, kemudian head akan menunjuk pada data baru tersebut sehingga head akan tetap selalu menjadi data terdepan. Untuk menghubungkan node terakhir dengan node terdepan dibutuhkan pointer bantu. Berikut ini contoh syntax untuk menambahkan node atau data di depan.

```
void insertDepan (int databaru)
{
    .....
    .....
}
```

5. Penambahan Node di Belakang

Penambahan node di belakang akan selalu dikaitkan dengan tail dan kemudian node baru tersebut akan menjadi tail. Berikut ini contoh syntax untuk menambahkan node dari belakang.

Syntax :

```
void insertBelakang(int databaru)
{
.....
.....
}
```

PEMBUATAN DOUBLE LINKED LIST CIRCULAR

Deklarasi node

Dibuat dari struct berikut ini:

```
typedef struct TNode{
    int data;
    TNode *next;
    TNode *prev;
};
```

Penjelasan:

- Pembuatan struct bernama TNode yang berisi 3 field, yaitu field databertipe integer dan field next & prev yang bertipe pointer dari TNode
- Setelah pembuatan struct, buat variabel head yang bertipe pointer dari TNode yang berguna sebagai kepala linked list.

Pembentukan node baru

Digunakan keyword new yang berarti mempersiapkan sebuah node baru beserta alokasi memorinya, pointer prev dan next menunjuk ke dirinya sendiri.

```
TNode *baru;
baru = new TNode;
baru->data = databaru;
baru->next = baru;
baru->prev = baru;
```

DOUBLE LINKED LIST CIRCULAR MENGGUNAKAN HEAD

- Menggunakan 1 pointer head
- Head selalu menunjuk node pertama

Deklarasi Pointer Head

Manipulasi linked list tidak bisa dilakukan langsung ke node yang dituju, melainkan harus melalui node pertama dalam linked list. Deklarasinya sebagai berikut:

```
TNode *head;
```

Fungsi Inisialisasi LinkedList

```
void init() {
    head = NULL;
}
```

Function untuk mengetahui kosong tidaknya LinkedList

```
Int isEmpty() {
    if(head == NULL) return 1;
    else return 0 ;
}
```

Contoh Circular Linked List :

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
```

```
struct node{
    int data;
    struct node *next;
};
```

```
typedef struct node node;
```

```
node *pList = NULL;
```

```
node *alokasiNodeBaru() {
    node *pNew = NULL;
    pNew = (node *)
    malloc(sizeof(node));
    return (pNew);
}
```

```
void tambahAwal(node *pNew) {
    node *temp;
    if(pList == NULL) {
```

```
pNew->next =
pNew;
    pList = pNew;
}
else{
    //cari node yang
    menunjuk ke pList
    temp = pList;
    while (temp->next
    != pList){
        temp =
temp->next;
    }

    temp->next =
pNew;
    pNew->next =
pList;
    pList = pNew;
}
}
```

```
void cetak(){
    node *pWalker = pList;
    node *pNext = NULL;
    while(pNext != pList){
        printf("pWalker = %d,
", pWalker->data);
        pNext = pWalker->next;
        pWalker = pNext;
        printf("pWalker->next
= %d \n", pWalker->data);
    }
}
```

```
void tambahTengah(node *pNew){
    node *pWalker;
    pWalker=pList;
```

```
int nilai, sisip;
    printf("masukkan nilai
yang ingin ditambahkan: ");
    scanf("%d", &nilai);
    pNew->data = nilai;
    cetak();
    printf("data disisipkan
setelah nilai : ");
    scanf("%d", &sisip);
    while(pWalker!=NULL &&
pWalker->data!=sisip){
        pWalker=pWalker->next; }

    if(pWalker->next==pList)
    printf("\ndata tidak
ditemukan");
    else {pNew->next=pWalker-
>next;
        pWalker->next=pNew;
    }
    cetak();
}
```

```
void tambahAkhir(node *pNew){
    node *pPre;
    pPre=pList;
    int nilai;
    printf("masukkan nilai
yang ingin ditambahkan: ");
    scanf("%d", &nilai);
    pNew->data = nilai;
    while(pPre->next!=pList){
        pPre=pPre->next; }
    pNew->next=pList;
    pPre->next=pNew;
}
```

```
void hapusAwal(){
    node *pEnd, *pHapus;
    pEnd=pList;
    pHapus=pList;
```

```
while (pEnd->next!=pList) {
    pEnd=pEnd->next;
    pEnd->next=pHapus->next;
    pList=pList->next;
    free (pHapus);
}

void hapusTengah() {
    node *pCari,*pPre;int
hapus;
    pPre=pList;
    pCari=pList;
    cetak();
    printf("masukkan bilangan
yang ingin dihapus: ");
    scanf("%d",&hapus);
    while (pCari->data!=hapus) {

        pCari=pCari->next; }
        while (pPre->next!=pCari) {

            pPre=pPre->next;
            pPre->next=pCari->next;
            free (pCari);
        }

    void hapusAkhir() {
        node *pEnd,*pPre;
        pEnd=pList;
        pPre=pList;
        while (pEnd->next!=pList) {

            pEnd=pEnd->next;
            while (pPre->next!=pEnd) {

                pPre=pPre->next;
                pPre->next=pList;
                free (pEnd);
            }

        }

        int main(int argc, char
        *argv[])
        {
```

```
        node *pNew; int pilih,bil;
        do{system("cls");
            printf("-----MENU-----
            -----");
            printf("\n1.tambah awal");
            printf("\n2.tambah
            tengah");
            printf("\n3.tambah akhir");
            printf("\n4.hapus awal ");
            printf("\n5.hapus tengah");
            printf("\n6.hapus akhir");
            printf("\n7.cetak");
            printf("\n8.exit");
            printf("\npilihan :
            ");scanf("%d",&pilih);

            if (pilih==1) {pNew=alokasiNodeBa
            ru();

                printf("masukkan bilangan: ");

                scanf("%d",&bil);

                pNew->data=bil;

                tambahAwal (pNew);
            }

            else
            if (pilih==2) {pNew=alokasiNodeBa
            ru();

                tambahTengah (pNew);
            }

            else
            if (pilih==3) {pNew=alokasiNodeBa
            ru();

                tambahAkhir (pNew);
            }

            else
            if (pilih==4) {hapusAwal ();}
            else
            if (pilih==5) {hapusTengah ();}
            else
            if (pilih==6) {hapusAkhir ();}
            else
            if (pilih==7) {cetak ();getch ();}
```

```
}while (pilih!=8) ;

/*
cetak() ;
pNew = alokasiNodeBaru() ;
tambahTengah(pNew) ;
*/
printf("\n") ;
system("PAUSE") ;
return 0 ;
}
galangcahya di 20.49
Berbagi
```

C. ID SECURITY

QWTD4452377-ASP-5244107

D. DISKUSI

Apakah artikel ini membantu anda?

KOMANG:Menurut saya artikel ini sangat membantu, karena dengan artikel ini saya bisa mengetahui tentang double linked list sircular dan bisa menambah pengetahuan saya.

Apa kekurangan artikel ini?

DIKI:Menurut saya kekurangan artikel ini adalah kurang nya gambar.

Apa kelebihan artikel ini?

BAGAS:Menurut saya kelebihan artikel ini terdapat pada penjelasannya yang sangat singkat dan jelas, dan terdapat contohnya.

E. KESIMPULAN

Setelah membuat artikel ini , menjadi lebih mengerti apa itu “Double Linked List Circular”

Double Linked List Circular pointer next dan prev nya menunjuk ke dirinya sendiri secara circular. Setiap node pada linked list mempunyai field yang berisi data dan pointer ke node berikutnya & ke node sebelumnya. Dibutuhkan dua buah variabel pointer yaitu head dan tail. Head akan selalu menunjuk pada node pertama, sedangkan tail akan selalu menunjuk pada node terakhir.

F. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," J. Inform., vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra, "COMPREHENSIVE SET OF PROFESSIONAL FOR DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU- POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."
- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra, "IMPLEMENTATION OF PARADOX DBASE."
- [11] A. S. Putra, "IMPLEMENTATION OF TRADE SECRET CASE STUDY SAMSUNG MOBILE PHONE."
- [12] A. S. Putra, "IMPLEMENTATION PATENT FOR APPLICATION WEB BASED CASE STUDI WWW.PUBLIKLAMPUNG.COM."
- [13] A. S. Putra, "IMPLEMENTATION SYSTEM FIRST TO INVENT IN DIGITALLY INDUSTRY."
- [14] A. S. Putra, "MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0."
- [15] A. S. Putra, "PATENT AS RELEVAN SUPPORT RESEARCH."
- [16] A. S. Putra, "PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc."
- [17] A. S. Putra, "PATENT PROTECTION FOR APPLICATION INVENT."
- [18] A. S. Putra, "QUICK REPORT IN PROPERTY PROGRAMMING."
- [19] A. S. Putra, "REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK."
- [20] A. S. Putra, "REVIEW TRADEMARK PATENT FOR INDUSTRIAL TECHNOLOGY BASED 4.0."
- [21] A. S. Putra, "TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING."
- [22] A. S. Putra, "WORKING DIRECTORY SET FOR PARADOX 7."
- [23] A. S. Putra, "ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk."
- [24] A. S. Putra, D. R. Aryanti, and I. Hartati, "Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya)," in Prosiding Seminar Nasional Darmajaya, 2018, vol. 1, no. 1, pp. 85–97.
- [25] A. S. Putra and O. M. Febriani, "Knowledge Management Online Application in PDAM Lampung Province," in Prosiding International conference on Information Technology and Business (ICITB), 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, "Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung," SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data), vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K.

Zuhri, “Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, pp. 221–232.

[28] D. P. Sari, O. M. Febriani, and A. S. Putra, “Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya,” in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.

Fakultas Komputer

MEYKA YULIANTI

Fakultas Komputer

MEYKA YULIANTI

C. ID SECURITY

QWTD4452377-ASP-5244107

D. DISKUSI

Apakah artikel ini membantu anda?

RIO :Sangat membantu sekali karena saya ingin belajar tentang struktur data ,dan artikel ini mudah di pahami karena menggunakan bahasa yang mudah untuk di mengerti bagi

kalangan mahasiswa seperti saya.

Apakah kekurangan artikel ini?

ROBI :Mungkin tambahkan lagi sedikit tentang penjelasan struktur data dengan ditambahkan foto saat membuat program aplikasi.

Apakah kelebihan artikel ini?

VERY :Kelebihan artikel ini adalah penjelasnya sangat singkat padat dan jelas.

Apakah artikel ini layak di publikasi?

DION : menurut saya artikel ini layak jika di publiksaikan, karena membantu mahasiswa untuk mencari informasi tentang struktur data dengan mudah.

E. KESIMPULAN

Jadi apa yang sudah di jelaskan diatas adalah kesimpulanya . **Larik (Array)** Larik (Bahasa Inggris: array), dalam ilmu komputer, adalah suatu tipe data terstruktur yang dapat menyimpan banyak data dengan suatu nama yang sama dan menempati tempat di memori yang berurutan (kontinu) serta bertipe data sama pula. Larik dapat diakses berdasarkan indeksnya. Indeks larik umumnya

dimulai dari 0 dan ada pula yang dimulai dari angka bukan 0. Pengaksesan larik biasanya dibuat dengan menggunakan perulangan (looping). dan seperti contohnya:

Larik Satu Dimensi

Larik satu dimensi merupakan jenis larik dasar dan jenis larik yang paling sering digunakan, pemakaian larik satu dimensi terutama dipakai dalam tipe data string (terutama dalam bahasa Bahasa pemrograman C).

Larik Dua Dimensi

Larik dua dimensi merupakan tipe larik yang lain. Larik dua dimensi sering dipakai untuk merepresentasikan tabel dan matriks dalam pemrograman.

Ada dan 2 jenis-jenis struktur data, ada struktur data sederhana misalnya array sedangkan struktur data majemuk ada stack.

Dan ada beberapa bahasa pemrograman seperti pascal, C++, java, php dan masih banyak lagi. Dalam teknik pemrograman, struktur data berarti tata letak data yang berisi kolom-kolom data, baik itu kolom yang tampak oleh pengguna (user) atau pun kolom yang hanya digunakan untuk keperluan pemrograman yang tidak tampak oleh pengguna. Setiap baris dari kumpulan kolom-kolom tersebut dinamakan catatan (record). Lebar kolom untuk data dapat berubah dan bervariasi.

F. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra, "COMPREHENSIVE SET OF PROFESSIONAL FOR DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU-POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."
- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra, "IMPLEMENTATION OF PARADOX DBASE."
- [11] A. S. Putra, "IMPLEMENTATION OF TRADE SECRET CASE STUDY SAMSUNG MOBILE PHONE."
- [12] A. S. Putra,

- “IMPLEMENTATION PATENT FOR APPLICATION WEB BASED CASE STUDI WWW. PUBLIKLAMPUNG.COM.”
- [13] A. S. Putra, “IMPLEMENTATION SYSTEM FIRST TO INVENT IN DIGITALLY INDUSTRY.”
- [14] A. S. Putra, “MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0.”
- [15] A. S. Putra, “PATENT AS RELEVAN SUPPORT RESEARCH.”
- [16] A. S. Putra, “PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc.”
- [17] A. S. Putra, “PATENT PROTECTION FOR APPLICATION INVENT.”
- [18] A. S. Putra, “QUICK REPORT IN PROPERTY PROGRAMMING.”
- [19] A. S. Putra, “REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK.”
- [20] A. S. Putra, “REVIEW TRADEMARK PATENT FOR INDUSTRIAL TECHNOLOGY BASED 4.0.”
- [21] A. S. Putra, “TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING.”
- [22] A. S. Putra, “WORKING DIRECTORY SET FOR PARADOX 7.”
- [23] A. S. Putra, “ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk.”
- [24] A. S. Putra, D. R. Aryanti, and I. Hartati, “Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya),” in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 85–97.
- [25] A. S. Putra and O. M. Febriani, “Knowledge Management Online Application in PDAM Lampung Province,” in *Prosiding International conference on Information Technology and Business (ICITB)*, 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, “Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung,” *SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data)*, vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K. Zuhri, “Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel,” *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 8, no. 2, pp. 221–232.
- [28] D. P. Sari, O. M. Febriani, and A. S. Putra, “Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya,” in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.