**LAPORAN *FINAL PROJECT***

**PEMROGRAMAN LANJUT**

****

**Nama Kelompok :**

**AVAIL WALAD [18081010151]**

**REYNATO YOURY FERRARY [19081010078]**

**SULTAN AGUNG NUGRAH ADI K. [20081010228]**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”**

**JAWA TIMUR**

**2021**

# LATAR BELAKANG

Rental Motor adalah suatu usaha yang bergerak di bidang jasa persewaan sepeda motor. pada situasi lingkungan sekitarnya yang mendukung usaha tersebut. Cara penyewaan motor dapat dipinjam secara harian ataupun kontrak mingguan ataupun bulanan. Pasar rental motor Indonesia saat ini terus meningkat karena dinilai lebih efisien, baik untuk kegiatan bisnis maupun keperluan pribadi yang bersifat temporer.

# ABSTRACT DATA TYPE

Struct adalah tipe data yang berisi kumpulan variable variable yang berada dalam satu nama yang sama dan memiliki keterkaitan beda dengan array jika array hanya sekumpulan variable yang bertipe data sama sedangkan struct bisa memiliki variable yang bertipe data berbeda contoh pada struct diatas terdapat integer dan char 2 tipe data yang berbeda

## SHELL SORT

Algoritma shell sort di temukan oleh Donald shell pada tahun 1959 algoritma ini disebut juga algoritma penambahan menurun. Metode ini mengurutkan data dengan cara membandingkan suatu data dengan data yang lain memiliki jarak tertentu sehingga membentuk sebuah sub list kemudian dilakukan penukaran jika diperlukan. Jarak yang di pakai cara menentukannya dengan rumus lalu sampai hasil angka tersebut menjadi 1

Contoh

Terdapat 5 buah bilangan berarti (dibulatkan kebawah)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3** | **4** | **5** | **1** | **2** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 1 | 2 | 4 | 5 |

4 dan 1 tukar

3 dan 1,2 tukar karena sudah memasuki jarak

5 dan 2 tukar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |

## JUMP SEARCH

Pada algoritma jump search, diperlukan data yang urut. Jarak lompatan optimal pada jump search adalah , dimana n adalah jumlah data. Lompatan dimulai dari nilai terkecil ke nilai terbesar. Setelah menemukan data yang lebih besar dari data yang dicari, maka hal selanjutnya yang dilakukan adalah melakukan pencarian linear untuk menentukan posisi data yang dicari.

# IMPLEMENTASI KODE

|  |
| --- |
| struct Sewa {  int id;  char nopol[12];  char tipe[36];  int tglsewa; // DD (01-31)  int blnsewa; // MM (01-12)  int thnsewa; // YYYY (2020)  int durasi;  char namapenyewa[36];  int harga;  } sewa[50], temp;  int jumlahSewa = 0; |

Ini adalah struct adalah tipe data yang berisi kumpulan variable variable yang berada dalam satu nama yang sama dan memiliki keterkaitan beda dengan array jika array hanya sekumpulan variable yang bertipe data sama sedangkan struct bisa memiliki variable yang bertipe data berbeda contoh pada struct diatas terdapat integer dan char 2 tipe data yang berbeda

|  |
| --- |
| bool edit(char nopol[]) {  int i;  for(i = 0; i<jumlahSewa; i++) {  if(!strcmp(nopol, sewa[i].nopol)) {  printf("Masukkan nama penyewa (saat ini: %s): ", sewa[i].namapenyewa);  scanf(" %[^\n]s", &sewa[i].namapenyewa);  printf("Masukkan tipe motor (saat ini: %s): ", sewa[i].tipe);  scanf(" %[^\n]s", &sewa[i].tipe);  printf("Masukkan nomor polisi (saat ini: %s): ", sewa[i].nopol);  scanf(" %[^\n]s", &sewa[i].nopol);  printf("Masukkan tanggal sewa (DD) (saat ini: %d): ", sewa[i].tglsewa);  scanf(" %d", &sewa[i].tglsewa);  printf("Masukkan bulan sewa (MM) (saat ini: %d): ", sewa[i].blnsewa);  scanf(" %d", &sewa[i].blnsewa);  printf("Masukkan tahun sewa (YYYY) (saat ini: %d): ", sewa[i].thnsewa);  scanf(" %d", &sewa[i].thnsewa);  printf("Masukkan durasi sewa (dalam hari) (saat ini: %d): ", sewa[i].durasi);  scanf(" %d", &sewa[i].durasi);  if(!strcmp(sewa[i].tipe, "Beat 125 CC")) sewa[i].harga = 100000\*sewa[i].durasi;  if(!strcmp(sewa[i].tipe, "Vario 150 CC")) sewa[i].harga = 150000\*sewa[i].durasi;  if(!strcmp(sewa[i].tipe, "Aerox")) sewa[i].harga = 200000\*sewa[i].durasi;  printf("Harga sewa: %d\n", sewa[i].harga);  }  }  return false;  } |

Ini adalah fungsi edit disini setelah memasukkan nopol yang akan di edit (source code terdapat pada menu utama) akan ada pertanyaan dengan mencantumkan data yang lama.

|  |
| --- |
| bool hapus(char nopol[]) {  int i;  for(i = 0; i<jumlahSewa; i++) {  if(!strcmp(nopol, sewa[i].nopol)) {  strcpy(sewa[i].nopol, " ");  return true;  }  }  return false;  } |

Ini adalah fungsi untuk menghapus data yang tersimpan pada struct disini untuk menghapus data harus memasukkan nopol yang ada pada menu utama

|  |
| --- |
| void tampilkan(int mode, int indeks) {  if(mode == 0) {  int i;  int out = 0;  for(i = 0; i<jumlahSewa; i++) {  if(!strcmp(" ", sewa[i].nopol)) continue;  else {  printf("\n[%d]", sewa[i].id);  printf("\nNama penyewa\t: %s", sewa[i].namapenyewa);  printf("\nTipe motor\t: %s", sewa[i].tipe);  printf("\nNomor polisi\t: %s", sewa[i].nopol);  printf("\nTanggal sewa\t: %d-%d-%d", sewa[i].tglsewa, sewa[i].blnsewa, sewa[i].thnsewa);  printf("\nDurasi sewa\t: %d hari", sewa[i].durasi);  printf("\nHarga sewa\t: Rp%d\n", sewa[i].harga);  out++;  }  if(out == 0) printf("\nTidak ada data persewaan yang ditambahkan!\n");  }  } else {  printf("\n[%d]", sewa[indeks].id);  printf("\nNama penyewa\t: %s", sewa[indeks].namapenyewa);  printf("\nTipe motor\t: %s", sewa[indeks].tipe);  printf("\nNomor polisi\t: %s", sewa[indeks].nopol);  printf("\nTanggal sewa\t: %d-%d-%d", sewa[indeks].tglsewa, sewa[indeks].blnsewa, sewa[indeks].thnsewa);  printf("\nDurasi sewa\t: %d hari", sewa[indeks].durasi);  printf("\nHarga sewa\t: Rp%d\n", sewa[indeks].harga);  }  } |

Ini adalah fungsi untuk menampilkan disini menggunakan for untuk menampilkan semua data yang ada pada struct. Disini juga menggunakan if else jika pada program ternyata belum ada data penyewa atau belum ada dalam struct.

|  |
| --- |
| bool baru() {  printf("Masukkan nama penyewa: ");  scanf(" %[^\n]s", &sewa[jumlahSewa].namapenyewa);  printf("Masukkan tipe motor: ");  scanf(" %[^\n]s", &sewa[jumlahSewa].tipe);  printf("Masukkan nomor polisi: ");  scanf(" %[^\n]s", &sewa[jumlahSewa].nopol);  printf("Masukkan tanggal sewa (DD): ");  scanf(" %d", &sewa[jumlahSewa].tglsewa);  printf("Masukkan bulan sewa (MM): ");  scanf(" %d", &sewa[jumlahSewa].blnsewa);  printf("Masukkan tahun sewa (YYYY): ");  scanf(" %d", &sewa[jumlahSewa].thnsewa);  printf("Masukkan durasi sewa (dalam hari): ");  scanf(" %d", &sewa[jumlahSewa].durasi);  if(!strcmp(sewa[jumlahSewa].tipe, "Beat 125 CC")) sewa[jumlahSewa].harga = 300000\*sewa[jumlahSewa].durasi;  if(!strcmp(sewa[jumlahSewa].tipe, "Vario 150 CC")) sewa[jumlahSewa].harga = 350000\*sewa[jumlahSewa].durasi;  if(!strcmp(sewa[jumlahSewa].tipe, "Aerox")) sewa[jumlahSewa].harga = 400000\*sewa[jumlahSewa].durasi;  printf("Harga sewa: %d\n", sewa[jumlahSewa].harga);  sewa[jumlahSewa].id = jumlahSewa+1;  return true;  } |

Ini adalah fungsi untuk input data persewaan baru disini user di minta untuk menginput nama jenis motor tanggal bulan tahun sewa, durasi sewa dan harga yang harus di keluarkan untuk menentukan harga hanya di kali antara durasi lama sewa dan harga dari tipe motor yang di pilih

|  |
| --- |
| void sorting(int mode){  int m, i, j;  if(mode == 1) { //ascending  for(m=jumlahSewa/2;m>0;m/=2)  {  for(j=m;j<jumlahSewa;j++)  {  for(i=j-m;i>=0;i-=m)  {  if(strcmp(sewa[i+m].nopol,sewa[i].nopol)>0)  break;  else  {  temp = sewa[i];  sewa[i] = sewa[i+m];  sewa[i+m] = temp;  }  }  }  }  } else if(mode == 2) { // descending  for(m=jumlahSewa/2;m>0;m/=2)  {  for(j=m;j<jumlahSewa;j++)  {  for(i=j-m;i>=0;i-=m)  {  if(strcmp(sewa[i+m].nopol,sewa[i].nopol)<0)  break;  else  {  temp = sewa[i];  sewa[i] = sewa[i+m];  sewa[i+m] = temp;  }  }  }  }  }    } |

Ini adalah fungsi untuk sorting, yaitu shell sort yang tadi sudah di jelaskan algoritmanya disini menggunakan nopol untuk sortingnya disini mengurutkannya secara ascending dari yang kecil ke besar atau discending dari yang besar ke kecil

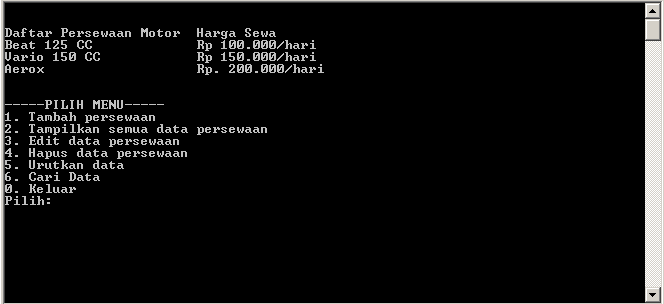
|  |
| --- |
| int jumpSrc(char nopol[], int n) {    sorting();  int jump = sqrt(n);  int previous = 0;    while (strcmp(sewa[(int)fmin(jump, n)-1].nopol, nopol) < 0) {  previous = jump;  jump += sqrt(n);  if (previous >= n) return -1;  }    while (strcmp(sewa[previous].nopol, nopol) < 0) {  previous++;  if (previous == (int)fmin(jump, n)) return -1;  }  if (!strcmp(sewa[previous].nopol, nopol)) return previous;  return -1;  } |

Ini adalah fungsi untuk melakukan pencarian data menggunakan jump search. Pada fungsi tersebut, kita mengambil parameter char nopol[] sebagai data yang dicari, dan parameter n sebagai jumlah data. Pada jump search, data yang dicari harus dalam keadaan urut secara ascending. Untuk mengurutkan data, dipanggil fungsi sorting() yang memanggil fungsi shell sort. Kemudian, kita menentukan jarak lompatan (jump), yaitu akar dari n. Apabila nilai dari sewa[angka terkecil dari jump atau n - 1].nopol lebih kecil dari input yang dicari, maka nilai previous adalah jump, dan jump ditambah dengan akar n. Jika previous lebih besar atau sama dengan n, maka data tidak ditemukan. Setelah perulangan pertama selesai, pada perulangan kedua, apabila nilai dari sewa[previous].nopol lebih kecil dari input yang dicari, maka previous akan terus ditambah. Jika previous sama dengan nilai jump atau jumlah data (n), maka data tidak ditemukan. Setelah perulangan selesai, maka didapat nilai dari indeks array yang dicari pada variabel previous apabila data tersebut ada. Keluaran dari fungsi ini akan diproses pada fungsi main, untuk kemudian dipanggil fungsi tampilkan() dengan argumen indeks array yang ingin ditampilkan.

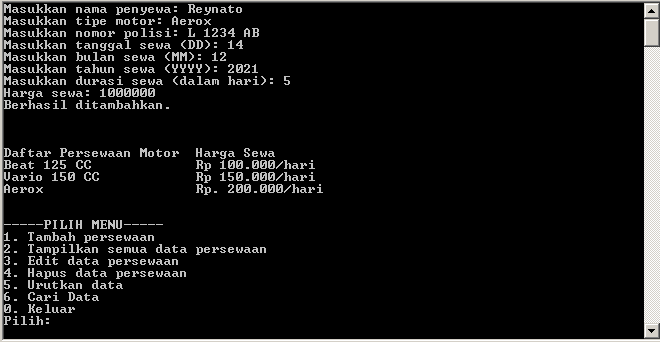
|  |
| --- |
| int main() {  //Menu  int pilih;  int i, sortmode;  char input[20];  bool run = true;  char inputNopol[12];  while(run == true) {  printf("\n\nDaftar Persewaan Motor\t\tHarga Sewa \n");  printf("Beat 125 CC\t\t\tRp 100.000/hari \n");  printf("Vario 150 CC\t\t\tRp 150.000/hari \n");  printf("Aerox\t\t\tRp 200.000/hari \n\n");  printf("\n-----PILIH MENU-----\n");  printf("1. Tambah persewaan\n2. Tampilkan semua data persewaan\n3. Edit data persewaan\n4. Hapus data persewaan\n5. Urutkan data\n6. Cari Data\n0. Keluar\nPilih: ");  scanf("%d", &pilih);    switch(pilih){  case 1:  system("cls");  if(input > 0) {  if(baru() == true) {  printf("Berhasil ditambahkan.\n\n");  jumlahSewa++;  } else printf("Gagal menambahkan. \n\n");  } else printf("Input tidak valid \n\n");  break;  case 2:  system("cls");  if(jumlahSewa > 0) tampilkan(0, 0);  break;  case 3:  system("cls");  printf("Masukkan nomor polisi dari motor yang akan diedit:\n");  scanf(" %[^\n]s", &inputNopol);  edit(inputNopol);  break;  case 4:  system("cls");  printf("Masukkan nomor polisi dari motor yang akan dihapus:\n");  scanf(" %[^\n]s", &inputNopol);  if(hapus(inputNopol) == true) printf("\nData dengan nomor polisi %s berhasil dihapus! \n\n", inputNopol);  else printf("Tidak dapat menemukan data dengan nomor polisi %s!", inputNopol);  break;  case 5:  system("cls");  printf("Masukkan mode pengurutan:\n1. Ascending\n2. Descending\n> ");  scanf(" %d", &sortmode);  sorting(sortmode);  printf("\nBerhasil mengurutkan berdasarkan nomor polisi, Silahkan lihat di menu tampilkan semua data. \n\n");  break;  case 6:  system("cls");  printf("Masukkan nomor polisi dari kendaraan yang ingin dicari: ");  scanf(" %[^\n]s", &inputNopol);  int cari = jumpSrc(inputNopol, jumlahSewa);  if(cari != -1) {  printf("\nHasil pencarian untuk \"%s\": \n", inputNopol);  tampilkan(1, cari);  } else printf("\nTidak ditemukan data dengan nomor polisi \"%s\"\n", inputNopol);  break;  case 0:  return 0;  default:  continue;  }  }  } |

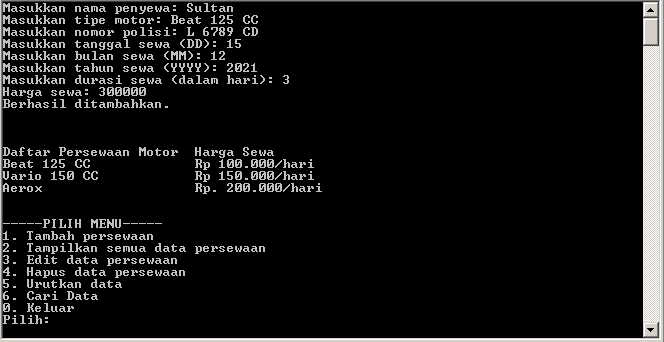
Ini adalah fungsi untuk menu utama menggunakan switch case disini menggunakan switch case karena jika sudah memilih 1 maka untuk case selanjutnya tidak perlu di cek

# SCREENSHOT APLIKASI

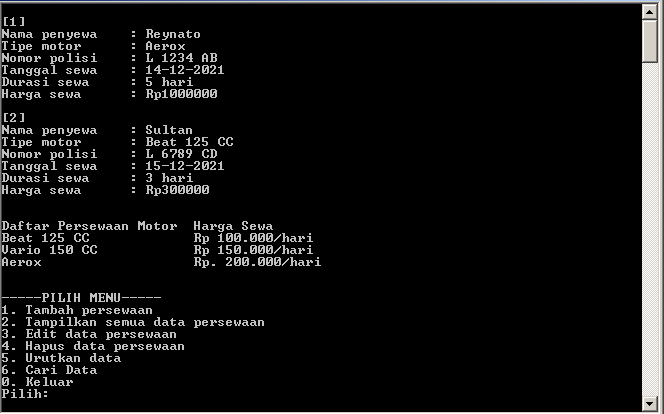


Menu Utama





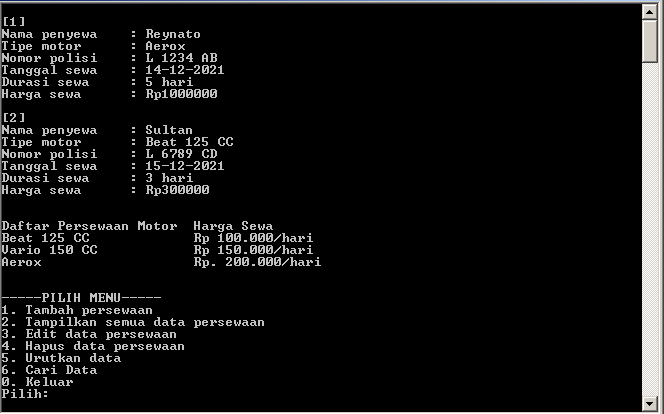
Input Data



Menu Tampilkan



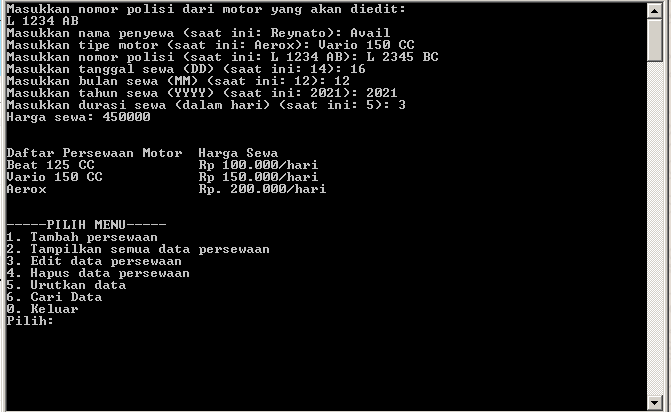
Menu Sort



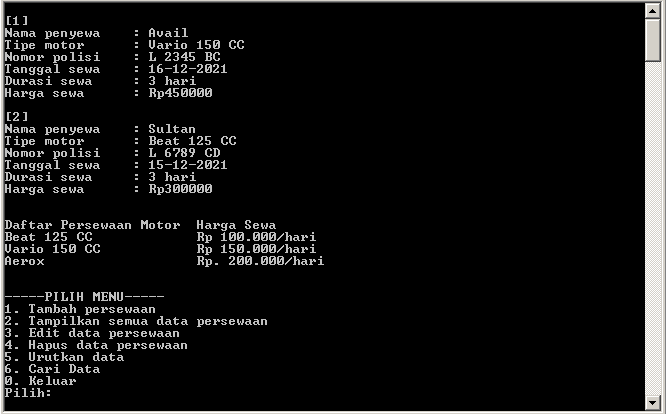
Menu Tampilkan Setelah Sorting



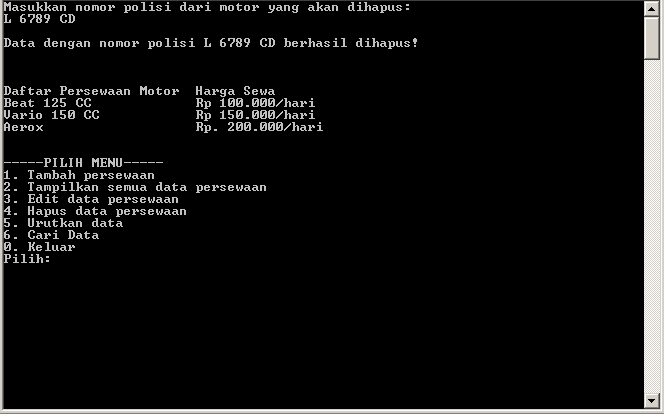
Menu Searching



Menu Edit



Menu Tampilkan Setelah Di Edit



Menu Hapus



Menu Tampilkan Setalah Di Hapus