**LAPORAN RESMI**

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

PRAKTIKUM KE: 2

MATERI: Linked List Insert (Int dan Struct)

DOSEN: Umi Sa’adah

NAMA MAHASISWA: Nadia Widad Naufalita (1 D4 IT A)

NRP: 2110151028

1. **PERCOBAAN INSERT (INT)**
2. **INSERT AWAL**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct simpul Node;

struct simpul{

int data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_awal();

void tampil();

main()

{

char pil;

printf("Percobaan Linked List Insert Awal\n");

printf("=================================\n");

do{

allocate\_node();

sisip\_awal();

fflush(stdin);

printf("Ingin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

}

void allocate\_node()

{

int nilai;

printf("\nData yang akan disimpan pada node baru: ");

scanf("%d", &nilai);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else

p->data = nilai;

}

void sisip\_awal()

{

p->next = head;

head = p;

}

void tampil()

{

baca = head;

puts(" ");

while(baca != NULL)

{

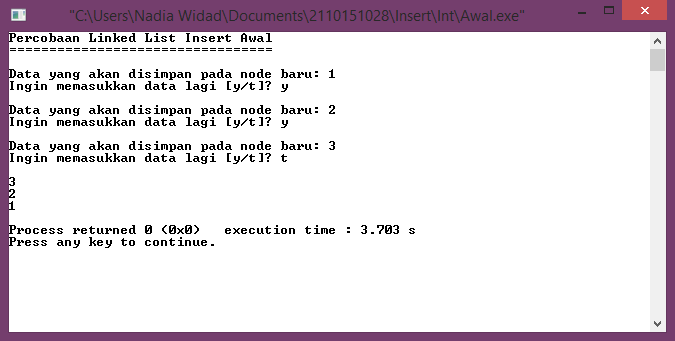
printf("%d\n", baca->data);

baca = baca->next;

}

}

* Capture Output:



1. **INSERT AKHIR**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct simpul Node;

struct simpul{

int data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_akhir();

void tampil();

main()

{

char pil;

printf("Percobaan Linked List Insert Akhir\n");

printf("==================================\n");

do{

allocate\_node();

sisip\_akhir();

fflush(stdin);

printf("Ingin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

}

void allocate\_node()

{

int nilai;

printf("\nData yang akan disimpan pada node baru: ");

scanf("%d", &nilai);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data = nilai;

p->next = NULL;

}

}

void sisip\_akhir()

{

Node \*tail;

if(head == NULL){

head = p;

}

else{

tail = head;

while(tail->next != NULL)

{

tail = tail->next;

}

tail->next = p;

tail = p;

}

}

void tampil()

{

baca = head;

puts(" ");

while(baca != NULL)

{

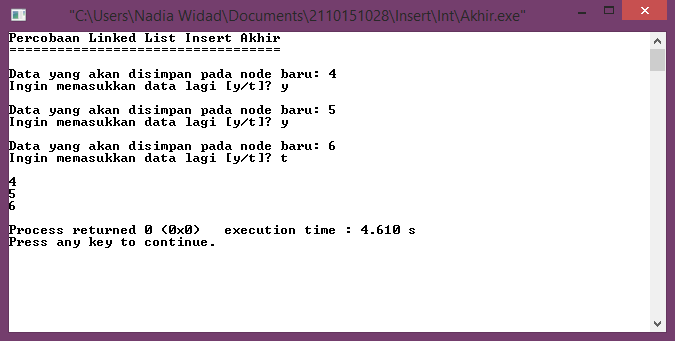
printf("%d\n", baca->data);

baca = baca->next;

}

}

* Capture Output:



1. **INSERT AFTER**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct simpul Node;

struct simpul{

int data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_akhir();

void sisip\_after(int);

void tampil();

main()

{

char pil;

int x;

printf("Percobaan Linked List Insert After\n");

printf("==================================\n");

do{

allocate\_node();

sisip\_akhir();

fflush(stdin);

printf("Ingin memasukkan data lagi? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

printf("\nData yang ingin dicari: ");

scanf("%d", &x);

allocate\_node();

sisip\_after(x);

tampil();

}

void allocate\_node()

{

int nilai;

printf("\nData yang akan disimpan pada node baru: ");

scanf("%d", &nilai);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data = nilai;

p->next = NULL;

}

}

void sisip\_akhir()

{

Node \*tail;

if(head == NULL){

head = p;

}

else{

tail = head;

while(tail->next != NULL)

{

tail = tail->next;

}

tail->next = p;

tail = p;

}

}

void sisip\_after(int x)

{

Node \*after;

if(head == NULL){

head = p;

}

else{

after = head;

while(after->data != x)

{

if(after->next == NULL){

printf("\n%d tidak ada!\n", x);

exit(0);

}

else{

after = after->next;

}

}

p->next = after->next;

after->next = p;

}

}

void tampil()

{

baca = head;

puts(" ");

while(baca != NULL)

{

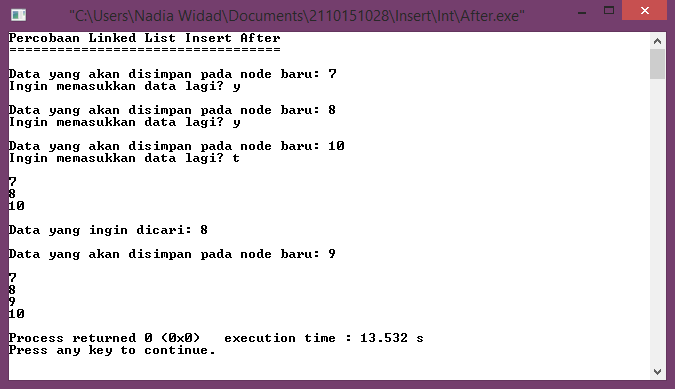
printf("%d\n", baca->data);

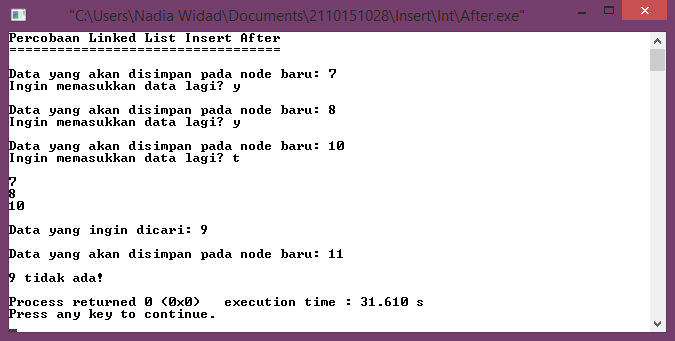
baca = baca->next;

}

}

* Capture Output:





1. **INSERT BEFORE**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct simpul Node;

struct simpul{

int data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_awal();

void sisip\_before(int);

void tampil();

main()

{

char pil;

int x;

printf("Percobaan Linked List Insert Before\n");

printf("===================================\n");

do{

allocate\_node();

sisip\_awal();

fflush(stdin);

printf("Ingin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

printf("\nData yang ingin dicari: ");

scanf("%d", &x);

allocate\_node();

sisip\_before(x);

tampil();

}

void allocate\_node()

{

int nilai;

printf("\nData yang akan disimpan pada node baru: ");

scanf("%d", &nilai);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data = nilai;

p->next = NULL;

}

}

void sisip\_awal()

{

p->next = head;

head = p;

}

void sisip\_before(int x)

{

Node \*bef;

Node \*prevbef;

if(head == NULL){

head = p;

}

else{

bef = head;

while(bef->data != x)

{

if(bef->next == NULL){

printf("\n%d tidak ada!\n", x);

exit(0);

}

else{

prevbef = bef;

bef = bef->next;

}

}

p->next = prevbef->next;

prevbef->next = p;

}

}

void tampil()

{

baca = head;

puts(" ");

while(baca != NULL)

{

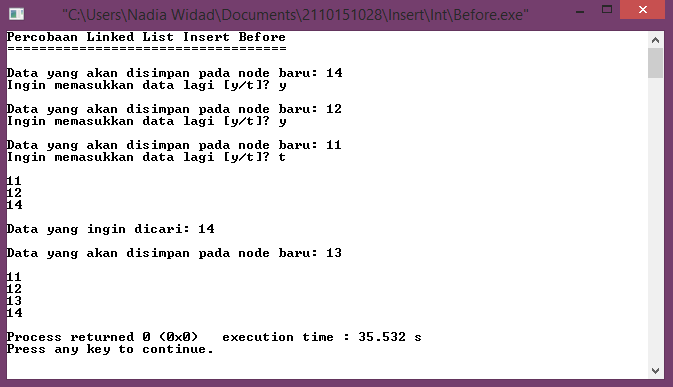
printf("%d\n", baca->data);

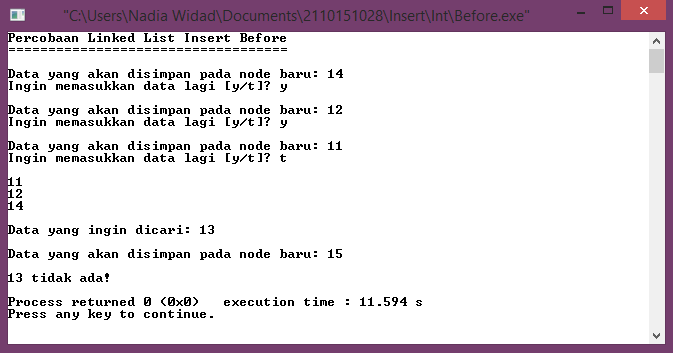
baca = baca->next;

}

}

* Capture Output:





1. **MENU INSERT**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct simpul Node;

struct simpul{

int data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_awal();

void sisip\_akhir();

void sisip\_after(int);

void sisip\_before(int);

void tampil();

main()

{

int menu, x;

char pil;

printf("Percobaan Linked List Menu Insert\n");

printf("=================================\n");

do{

printf("\nMENU INSERT\n");

printf("\n1. AWAL");

printf("\n2. AKHIR");

printf("\n3. AFTER");

printf("\n4. BEFORE\n");

printf("\nMasukkan pilihan: ");

scanf("%d", &menu);

switch(menu)

{

case 1:

allocate\_node();

sisip\_awal();

break;

case 2:

allocate\_node();

sisip\_akhir();

break;

case 3:

allocate\_node();

printf("\nData yang ingin dicari: ");

scanf("%d", &x);

sisip\_after(x);

break;

case 4:

allocate\_node();

printf("\nData yang ingin dicari: ");

scanf("%d", &x);

sisip\_before(x);

break;

default:

printf("\nPilihan tidak tersedia!");

break;

}

fflush(stdin);

printf("Ingin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

}

void allocate\_node()

{

int nilai;

printf("\nData yang akan disimpan pada node baru: ");

scanf("%d", &nilai);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else

p->data = nilai;

p->next = NULL;

}

void sisip\_awal()

{

p->next = head;

head = p;

}

void sisip\_akhir()

{

Node \*tail;

if(head == NULL){

head = p;

}

else{

tail = head;

while(tail->next != NULL)

{

tail = tail->next;

}

tail->next = p;

tail = p;

}

}

void sisip\_after(int x)

{

Node \*after;

if(head == NULL){

head = p;

}

else{

after = head;

while(after->data != x)

{

if(after->next == NULL){

printf("\n%d tidak ada!\n", x);

exit(0);

}

else{

after = after->next;

}

}

p->next = after->next;

after->next = p;

}

}

void sisip\_before(int x)

{

Node \*bef;

Node \*prevbef;

if(head == NULL){

head = p;

}

else{

bef = head;

while(bef->data != x)

{

if(bef->next == NULL){

printf("\n%d tidak ada!\n", x);

exit(0);

}

else{

prevbef = bef;

bef = bef->next;

}

}

p->next = prevbef->next;

prevbef->next = p;

}

}

void tampil()

{

baca = head;

puts(" ");

while(baca != NULL)

{

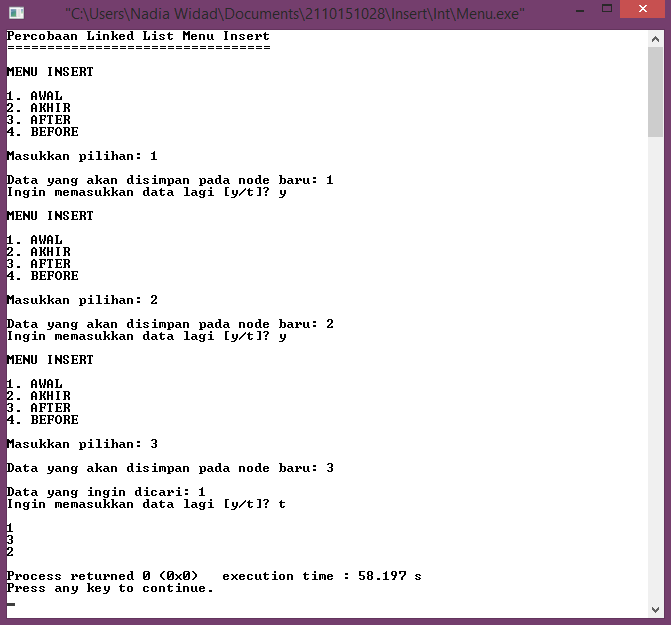
printf("%d\n", baca->data);

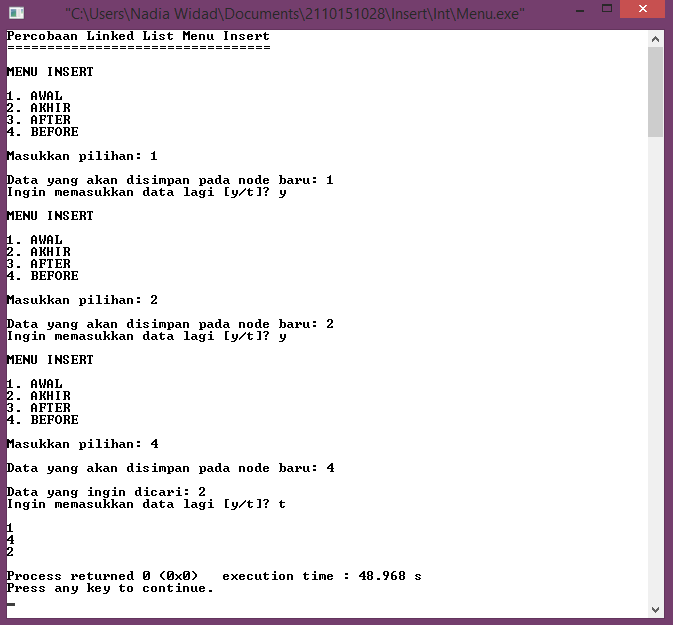
baca = baca->next;

}

}

* Capture Output:





1. **PERCOBAAN INSERT (STRUCT)**
2. **INSERT AWAL**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct{

int no, nrp;

char nama[50], jurusan[25];

float nilai;

} mahasiswa;

typedef struct simpul Node;

struct simpul{

mahasiswa data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_awal();

void tampil();

main()

{

char pil;

printf("Percobaan Linked List Insert Awal\n");

printf("=================================\n");

do{

allocate\_node();

sisip\_awal();

fflush(stdin);

printf("\nIngin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

}

void allocate\_node()

{

int nomor, NRP;

char name[50], jur[25];

float score;

printf("\nMasukkan Data\n");

printf("No.\t\t: ");

scanf("%d", &nomor);

printf("NRP\t\t: ");

scanf("%d", &NRP);

fflush(stdin);

printf("Nama\t\t: ");

scanf("%s", &name);

fflush(stdin);

printf("Jurusan\t\t: ");

scanf("%s", &jur);

printf("Nilai\t\t: ");

scanf("%f", &score);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data.no = nomor;

p->data.nrp = NRP;

strcpy(p->data.nama, name);

strcpy(p->data.jurusan, jur);

p->data.nilai = score;

}

}

void sisip\_awal()

{

p->next = head;

head = p;

}

void tampil()

{

baca = head;

printf("\n\t\t\tDATA MAHASISWA\n");

printf("-------------------------------------------------------------\n");

printf("No.\tNRP\t\tNama\t\tJurusan\t\tNilai\t\n");

printf("-------------------------------------------------------------\n");

while(baca != NULL)

{

printf("%d\t%d\t%s\t\t%s\t\t%.f\t\n", baca->data.no, baca->data.nrp, baca->data.nama, baca->data.jurusan, baca->data.nilai);

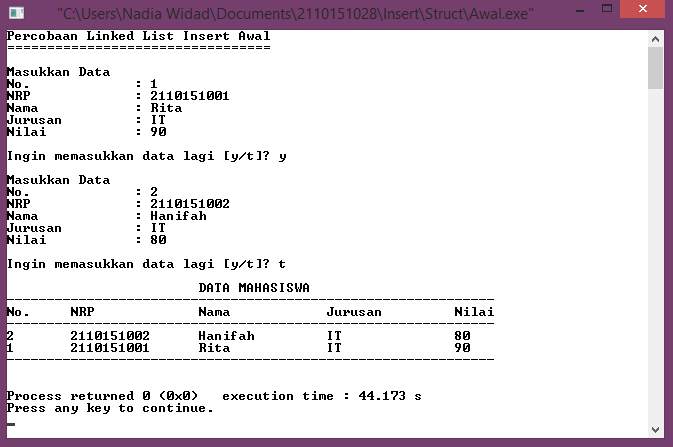
baca = baca->next;

}

printf("-------------------------------------------------------------\n\n");

}

* Capture Output:



1. **INSERT AKHIR**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct{

int no, nrp;

char nama[50], jurusan[25];

float nilai;

} mahasiswa;

typedef struct simpul Node;

struct simpul{

mahasiswa data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_akhir();

void tampil();

main()

{

char pil;

printf("Percobaan Linked List Insert Akhir\n");

printf("==================================\n");

do{

allocate\_node();

sisip\_akhir();

fflush(stdin);

printf("\nIngin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

}

void allocate\_node()

{

int nomor, NRP;

char name[50], jur[25];

float score;

printf("\nMasukkan Data\n");

printf("No.\t\t: ");

scanf("%d", &nomor);

printf("NRP\t\t: ");

scanf("%d", &NRP);

fflush(stdin);

printf("Nama\t\t: ");

scanf("%s", &name);

fflush(stdin);

printf("Jurusan\t\t: ");

scanf("%s", &jur);

printf("Nilai\t\t: ");

scanf("%f", &score);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data.no = nomor;

p->data.nrp = NRP;

strcpy(p->data.nama, name);

strcpy(p->data.jurusan, jur);

p->data.nilai = score;

p->next = NULL;

}

}

void sisip\_akhir()

{

Node \*tail;

if(head == NULL){

head = p;

}

else{

tail = head;

while(tail->next != NULL)

{

tail = tail->next;

}

tail->next = p;

tail = p;

}

}

void tampil()

{

baca = head;

printf("\n\t\t\tDATA MAHASISWA\n");

printf("-------------------------------------------------------------\n");

printf("No.\tNRP\t\tNama\t\tJurusan\t\tNilai\t\n");

printf("-------------------------------------------------------------\n");

while(baca != NULL)

{

printf("%d\t%d\t%s\t\t%s\t\t%.f\t\n", baca->data.no, baca->data.nrp, baca->data.nama, baca->data.jurusan, baca->data.nilai);

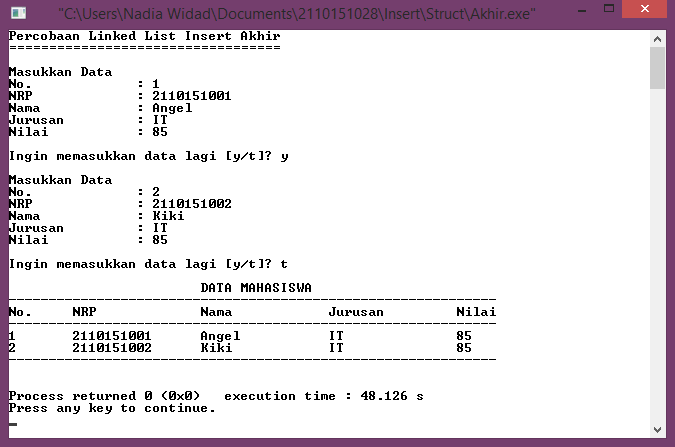
baca = baca->next;

}

printf("-------------------------------------------------------------\n\n");

}

* Capture Output:



1. **INSERT AFTER**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct{

int no, nrp;

char nama[50], jurusan[25];

float nilai;

} mahasiswa;

typedef struct simpul Node;

struct simpul{

mahasiswa data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_akhir();

void sisip\_after(int);

void tampil();

main()

{

char pil;

int x;

printf("Percobaan Linked List Insert After\n");

printf("==================================\n");

do{

allocate\_node();

sisip\_akhir();

fflush(stdin);

printf("\nIngin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

printf("\nNo. yang ingin dicari: ");

scanf("%d", &x);

allocate\_node();

sisip\_after(x);

tampil();

}

void allocate\_node()

{

int nomor, NRP;

char name[50], jur[25];

float score;

printf("\nMasukkan Data\n");

printf("No.\t\t: ");

scanf("%d", &nomor);

printf("NRP\t\t: ");

scanf("%d", &NRP);

fflush(stdin);

printf("Nama\t\t: ");

scanf("%s", &name);

fflush(stdin);

printf("Jurusan\t\t: ");

scanf("%s", &jur);

printf("Nilai\t\t: ");

scanf("%f", &score);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data.no = nomor;

p->data.nrp = NRP;

strcpy(p->data.nama, name);

strcpy(p->data.jurusan, jur);

p->data.nilai = score;

p->next = NULL;

}

}

void sisip\_akhir()

{

Node \*tail;

if(head == NULL){

head = p;

}

else{

tail = head;

while(tail->next !=NULL)

{

tail = tail->next;

}

tail->next = p;

tail = p;

}

}

void sisip\_after(int x)

{

Node \*after;

if(head == NULL){

head = p;

}

else{

after = head;

while(after->data.no != x)

{

if(after->next == NULL){

printf("\nData tidak ada!\n");

exit(0);

}

else{

after = after->next;

}

}

p->next = after->next;

after->next = p;

}

}

void tampil()

{

baca = head;

printf("\n\t\t\tDATA MAHASISWA\n");

printf("-------------------------------------------------------------\n");

printf("No.\tNRP\t\tNama\t\tJurusan\t\tNilai\t\n");

printf("-------------------------------------------------------------\n");

while(baca != NULL)

{

printf("%d\t%d\t%s\t\t%s\t\t%.f\t\n", baca->data.no, baca->data.nrp, baca->data.nama, baca->data.jurusan, baca->data.nilai);

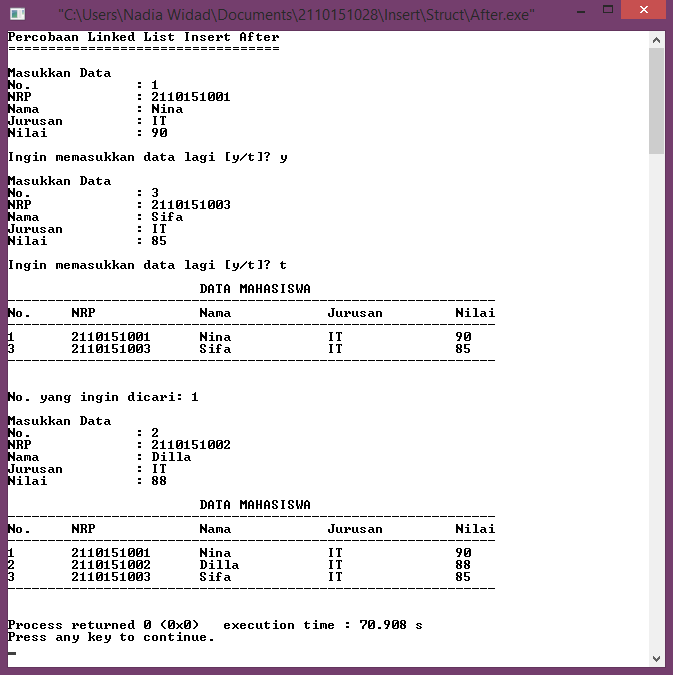
baca = baca->next;

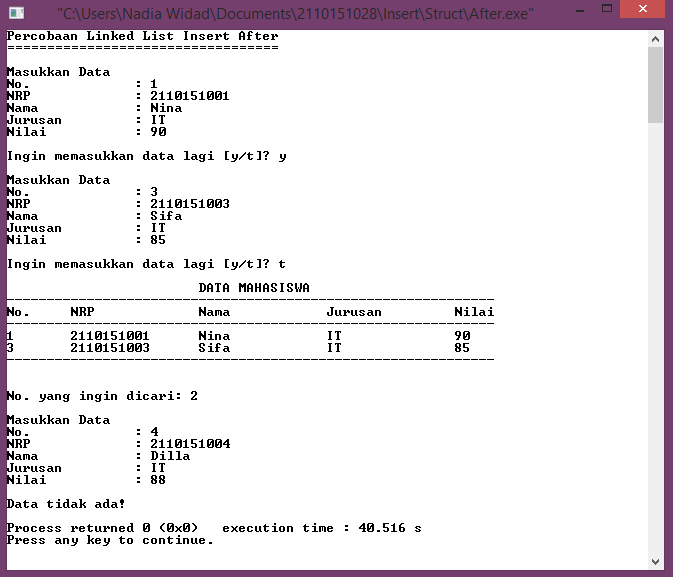
}

printf("-------------------------------------------------------------\n\n");

}

* Capture Output:





1. **INSERT BEFORE**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct{

int no, nrp;

char nama[50], jurusan[25];

float nilai;

} mahasiswa;

typedef struct simpul Node;

struct simpul{

mahasiswa data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_awal();

void sisip\_before(int);

void tampil();

main()

{

char pil;

int x;

printf("Percobaan Linked List Insert Before\n");

printf("===================================\n");

do{

allocate\_node();

sisip\_awal();

fflush(stdin);

printf("\nIngin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

printf("\nNo. yang ingin dicari: ");

scanf("%d", &x);

allocate\_node();

sisip\_before(x);

tampil();

}

void allocate\_node()

{

int nomor, NRP;

char name[50], jur[25];

float score;

printf("\nMasukkan Data\n");

printf("No.\t\t: ");

scanf("%d", &nomor);

printf("NRP\t\t: ");

scanf("%d", &NRP);

fflush(stdin);

printf("Nama\t\t: ");

scanf("%s", &name);

fflush(stdin);

printf("Jurusan\t\t: ");

scanf("%s", &jur);

printf("Nilai\t\t: ");

scanf("%f", &score);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data.no = nomor;

p->data.nrp = NRP;

strcpy(p->data.nama, name);

strcpy(p->data.jurusan, jur);

p->data.nilai = score;

p->next = NULL;

}

}

void sisip\_awal()

{

p->next = head;

head = p;

}

void sisip\_before(int x)

{

Node \*bef;

Node \*prevbef;

if(head == NULL){

head = p;

}

else{

bef = head;

while(bef->data.no != x)

{

if(bef->next == NULL){

printf("\nData tidak ada!\n");

exit(0);

}

else{

prevbef = bef;

bef = bef->next;

}

}

p->next = prevbef->next;

prevbef->next = p;

}

}

void tampil()

{

baca = head;

printf("\n\t\t\tDATA MAHASISWA\n");

printf("-------------------------------------------------------------\n");

printf("No.\tNRP\t\tNama\t\tJurusan\t\tNilai\t\n");

printf("-------------------------------------------------------------\n");

while(baca != NULL)

{

printf("%d\t%d\t%s\t\t%s\t\t%.f\t\n", baca->data.no, baca->data.nrp, baca->data.nama, baca->data.jurusan, baca->data.nilai);

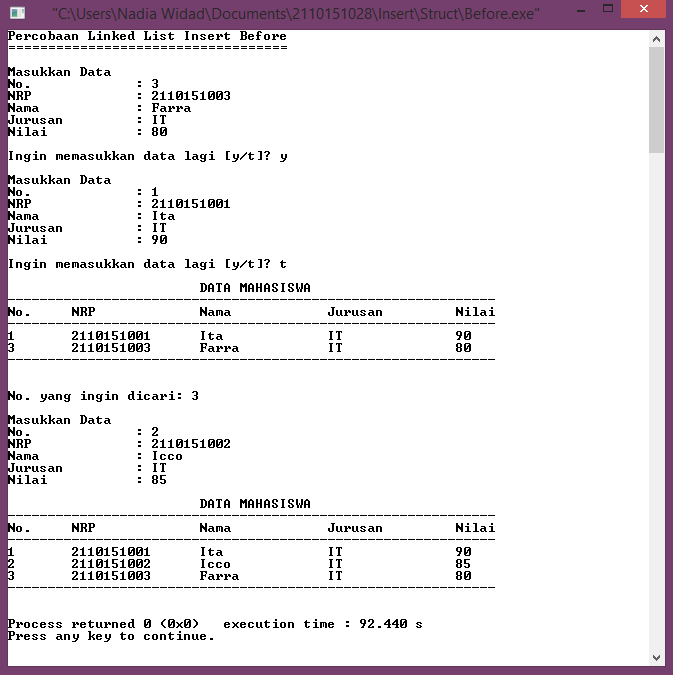
baca = baca->next;

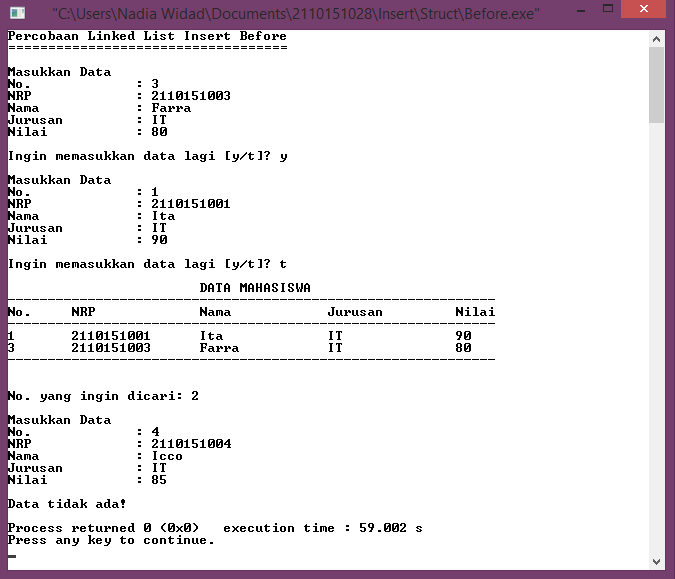
}

printf("-------------------------------------------------------------\n\n");

}

* Capture Output:





1. **MENU INSERT**

* Listing Program:

#include <stdio.h>

#include <stdlib.h>

typedef struct{

int no, nrp;

char nama[50], jurusan[25];

float nilai;

} mahasiswa;

typedef struct simpul Node;

struct simpul{

mahasiswa data;

Node \*next;

};

Node \*head = NULL;

Node \*p;

Node \*baca;

void allocate\_node();

void sisip\_awal();

void sisip\_akhir();

void sisip\_after(int);

void sisip\_before(int);

void tampil();

main()

{

int menu, x;

char pil;

printf("Percobaan Linked List Menu Insert\n");

printf("=================================\n");

do{

printf("\nMENU INSERT\n");

printf("\n1. AWAL");

printf("\n2. AKHIR");

printf("\n3. AFTER");

printf("\n4. BEFORE\n");

printf("\nMasukkan pilihan: ");

scanf("%d", &menu);

switch(menu)

{

case 1:

allocate\_node();

sisip\_awal();

break;

case 2:

allocate\_node();

sisip\_akhir();

break;

case 3:

allocate\_node();

printf("\nNo. yang ingin dicari: ");

scanf("%d", &x);

sisip\_after(x);

break;

case 4:

allocate\_node();

printf("\nNo. yang ingin dicari: ");

scanf("%d", &x);

sisip\_before(x);

break;

default:

printf("\nPilihan tidak tersedia!");

break;

}

fflush(stdin);

printf("\nIngin memasukkan data lagi [y/t]? ");

scanf("%c", &pil);

}while(pil == 'y' || pil == 'Y');

tampil();

}

void allocate\_node()

{

int nomor, NRP;

char name[50], jur[25];

float score;

printf("\nMasukkan Data\n");

printf("No.\t\t: ");

scanf("%d", &nomor);

printf("NRP\t\t: ");

scanf("%d", &NRP);

fflush(stdin);

printf("Nama\t\t: ");

scanf("%s", &name);

fflush(stdin);

printf("Jurusan\t\t: ");

scanf("%s", &jur);

printf("Nilai\t\t: ");

scanf("%f", &score);

p = (Node \*)malloc(sizeof(Node));

if(p == NULL)

exit(0);

else{

p->data.no = nomor;

p->data.nrp = NRP;

strcpy(p->data.nama, name);

strcpy(p->data.jurusan, jur);

p->data.nilai = score;

p->next = NULL;

}

}

void sisip\_awal()

{

p->next = head;

head = p;

}

void sisip\_akhir()

{

Node \*tail;

if(head == NULL){

head = p;

}

else{

tail = head;

while(tail->next != NULL)

{

tail = tail->next;

}

tail->next = p;

tail = p;

}

}

void sisip\_after(int x)

{

Node \*after;

if(head == NULL){

head = p;

}

else{

after = head;

while(after->data.no != x)

{

if(after->next == NULL){

printf("\nData tidak ada!\n");

exit(0);

}

else{

after = after->next;

}

}

p->next = after->next;

after->next = p;

}

}

void sisip\_before(int x)

{

Node \*bef;

Node \*prevbef;

if(head == NULL){

head = p;

}

else{

bef = head;

while(bef->data.no != x)

{

if(bef->next == NULL){

printf("\nData tidak ada!\n");

exit(0);

}

else{

prevbef = bef;

bef = bef->next;

}

}

p->next = prevbef->next;

prevbef->next = p;

}

}

void tampil()

{

baca = head;

printf("\n\t\t\tDATA MAHASISWA\n");

printf("-------------------------------------------------------------\n");

printf("No.\tNRP\t\tNama\t\tJurusan\t\tNilai\t\n");

printf("-------------------------------------------------------------\n");

while(baca != NULL)

{

printf("%d\t%d\t%s\t\t%s\t\t%.f\t\n", baca->data.no, baca->data.nrp, baca->data.nama, baca->data.jurusan, baca->data.nilai);

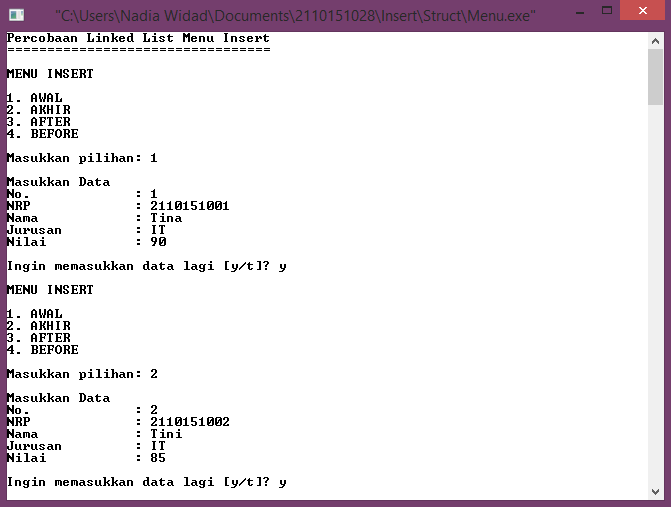
baca = baca->next;

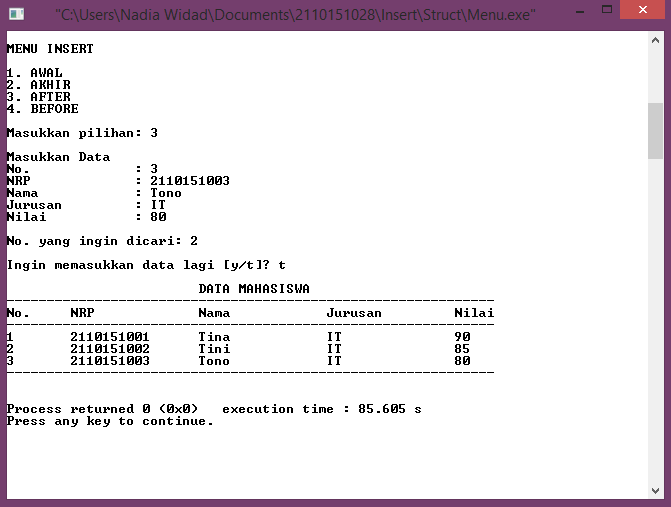
}

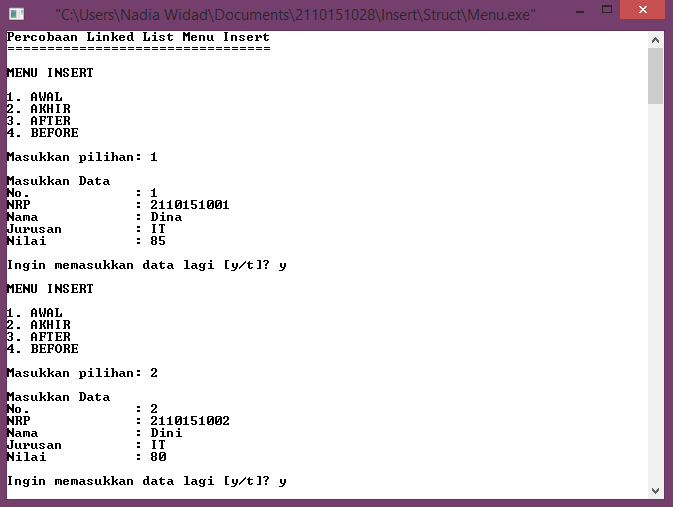
printf("-------------------------------------------------------------\n\n");

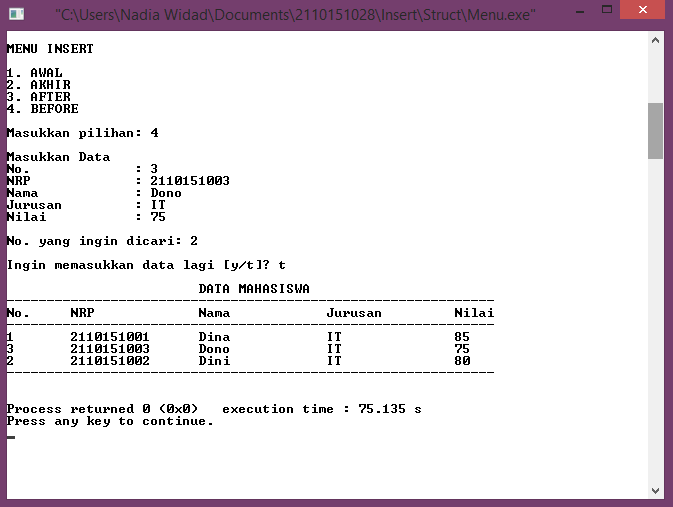
}

* Capture Output:









1. **KESIMPULAN**

* Linked list adalah kumpulan linier sejumlah data. Contoh implementasi linked list misalnya sebuah list yang berisi daftar belanjaan, daftar nilai mahasiswa, dll.
* Linked list terdiri dari elemen-elemen individu, dimana masing-masing dihubungkan dengan pointer tunggal.
* Masing-masing elemen terdiri dari dua bagian, yaitu bagian data/informasi yang disimpan dan bagian pointer yang disebut dengan pointer *next*.
* Pointer *next* pada elemen terakhir merupakan NULL, yang menunjukkan akhir dari suatu list.
* Elemen pada awal suatu list disebut *head*, dan elemen pada akhir suatu list disebut *tail*.
* Struktur node pada linked list merupakan suatu simpul (node) yang berisi pointer ke suatu data yang merupakan data dirinya sendiri.
* Untuk melakukan pemesanan memory berulang-ulang sesuai dengan kebutuhan dan mengurangi duplikasi kode, maka dibuat sebuah fungsi allocate\_node() menggunakan malloc() untuk mendapatkan memori aktual.
* Pada dasarnya, percobaan linked list insert dengan tipe data integer dan struct menggunakan cara yang sama. Hanya saja jika insert dengan tipe data integer, maka data yang disimpan pada node berupa bilangan integer, sedangkan insert dengan tipe data struct, data yang disimpan pada node berupa elemen-elemen dari struct tersebut yang memiliki berbagai macam tipe data.